

Приложение 2.1. към Техническото предложение

**Техническото предложение – Предложение за изпълнение на обществената поръчка с предмет:
„ Развитие на информационна система „Външна търговия“ с цел добавяне на нови функционалности и миграция на платформата“**

„Смарт Системс 2010“ ЕООД

2019

Съдържание:

1. РЕЧНИК НА ТЕРМИНИ, ДЕФИНИЦИИ И СЪКРАЩЕНИЯ	10
1.1. Използвани съкращения	10
1.2. Технологични дефиниции	12
1.3. Дефиниции за нива на електронизация на услугите	14
2. ПРЕДСТАВЯНЕ НА УЧАСТНИКА - СМАРТ СИСТЕМС 2010 ЕООД	15
3. ЗА ПРОЕКТА	16
3.1. Общи и специфични цели на проекта	16
3.2. Обхват на обществената поръчка	17
3.3. Целеви групи	17
3.4. Очаквани резултати	18
3.5. Период на изпълнение	18
4. Организация и методология за управление на проекта	18
4.1. Общи организационни принципи	18
4.2. Методология за изпълнение на проекта	20
4.2.1. Обосновка на използването на RUP за успешното изпълнение на проекта	22
4.2.2. Фази на проекта	25
4.2.3. Функционални особености на приложения подход – съотнесимост на методологията RUP към фазите на конкретния проект	25
4.2.4. Използвани концепции, методи и инструменти	28
4.2.5. Адаптиране на предлаганата методика към спецификата на проекта	29
4.2.6. Гарантиране и повишаване на качеството чрез предлаганите методи за изпълнение в предлаганата организация и методология за изпълнение на договора	31
4.2.7. Обосновка на предлаганата методика за изпълнение и предлаганата организация за изпълнение към реализирането на резултатите от обществената поръчка	33
4.3. Организация на екипа за изпълнение	34
4.3.1.1. Роли и отговорности на членовете на екипа на Сمارт Системс	43
4.3.1.2. Структура на екипа на Сمارт Системс	51
4.3.1.3. Начин на взаимодействие между членовете на екипа на Смарт Системс ;	51
4.3.1.4. Връзки за взаимодействие с екипа на Възложителя	52
4.4. Проектна документация	55
4.5. Управление на качеството	55
4.5.1. Дефиниция и цел	55

4.5.2.	Елементи на подхода за управление на качеството	58
4.5.3.	Измерване и осигуряване на качеството	59
4.5.4.	Мерки за мониторинг, контрол, измерване на качеството на изпълнение на дейностите и на персонала във фазите на проекта	59
4.5.5.	Приложимост на мерките за мониторинг и контрол към отделните фази на проекта	63
4.5.6.	Входни данни за осигуряване (измерване) на качеството	63
4.5.7.	Одити	64
4.5.8.	Натрупване на допълнително информация с цел анализ и бъдещи действия	64
4.5.9.	Извеждане на изводи и резултати от извършени одити	65
4.5.10.	Анализ на процесите	66
4.5.11.	Инструменти и методи на контролиране на качеството	66
4.5.12.	Валидиране на резултати и мерки за контрол на промяна.....	68
4.5.13.	Приложение на подхода за управление на качеството във фазите на проекта	70
4.5.14.	Използване на методи и подходи за вземане на решения, основани на действителни/реални факти и информация (Factual approach to decision making)	74
4.5.15.	Подход за изготвяне на план за управление на качеството	75
4.6.	График за изпълнение на проекта	77
4.7.	Управление на риска	79
4.7.1.	Планиране на управлението на риска	80
4.7.2.	Идентифициране на рисковете	81
4.7.3.	Анализ.....	82
4.7.3.1.	Качествен анализ (Приоритизация на рисковете).....	82
4.7.3.2.	Количествен анализ	84
4.7.3.3.	Планиране на реакцията.....	84
4.7.3.4.	Наблюдение и контрол	86
4.7.4.	Прилагане на Методологията за управление на риска на проекта	87
4.7.4.1.	Планиране на управлението на риска.....	87
4.7.4.2.	Статично измерение на дейността „Планиране на управлението на риска“	87
4.7.4.3.	Динамично измерение на дейността „Планиране на управлението на риска“	88
4.7.4.4.	Идентифициране на рисковете	88
4.7.4.5.	Анализ и Планиране на реакцията	94
4.7.4.6.	Наблюдение и контрол	97

5. ЕТАПИ НА РЕАЛИЗАЦИЯ НА ДЕЙНОСТИТЕ ПО ПРОЕКТА	99
5.1. Анализ на данните и изискванията	99
5.2. Изготвяне на системен проект	100
5.3. Разработване на софтуерното решение	100
5.4. Тестване.....	101
5.5. Внедряване.....	101
5.6. Обучение.....	101
5.7. Гаранционна поддръжка	102
6. СЪОТВЕТСТВИЕ С ОБЩИТЕ ИЗИСКВАНИЯ ЗА ИНФОРМАЦИОННИ СИСТЕМИ В ДЪРЖАВНАТА АДМИНИСТРАЦИЯ	104
6.1. Функционални изисквания	104
6.1.1. Интеграция с външни информационни системи	104
6.1.2. Интеграционен слой	104
6.1.3. Технически изисквания към интерфейсите.....	107
6.1.4. Електронна идентификация на потребителите	108
6.1.5. Формиране на изгледи.....	111
6.1.6. Администриране на Системата	112
6.2. Нефункционални изисквания към информационната система.....	112
6.2.1. Авторски права и изходен код	112
6.2.2. Системна и приложна архитектура.....	113
6.2.3. Повторно използване (преизползване) на ресурси и готови разработки	116
6.2.4. Изграждане и поддръжка на множество среди.....	117
6.2.5. Процес на разработка, тестване и разгръщане	118
6.2.6. Бързодействие и мащабируемост.....	119
6.2.6.1. Контрол на натоварването и защита от DoS/DDoS атаки	119
6.2.6.2. Бързодействие	119
6.2.6.3. Използване на HTTP/2.....	119
6.2.6.4. Качество и сигурност на програмните продукти и приложенията	122
6.2.6.5. Информационна сигурност и интегритет на данните	122
6.2.6.6. Използваемост.....	125
6.2.7. Системен журнал.....	128
6.2.8. Дизайн на бази данни и взаимодействие с тях	129
7. ПОДХОД ЗА ИЗПЪЛНЕНИЕ НА ИЗИСКВАНИЯТА КЪМ ПОРЪЧКАТА ..	129
7.1. Предложение за извършване на дейностите по анализ и проектиране	129
7.1.1. Методология за анализ на процесите и моделиране	130

7.1.1.1.	Методи, стъпки и инструменти за реализация на дейностите по анализ и проектиране.....	132
7.1.2.	Метод за спецификация на потребителските случаи.....	134
7.1.2.1.	Подход към осигуряване на качеството на продукта от дейностите по спецификация на системния проект.....	137
7.1.3.	Методика за проектиране.....	138
7.1.3.1.	Принципи за дизайн.....	139
7.1.3.2.	Шаблони за дизайн (Design patterns)	140
7.1.3.3.	Приложение на итеративния подход в обектно – ориентирано проектиране.....	140
7.1.3.4.	Методика за проектиране на потребителските интерфейси	143
7.1.4.	Инструменти за описание на аналитичните дейности.....	144
7.2.	Методики за извършване на дейностите по разработка на системата	148
7.2.1.	Методология за извършване на дейностите по разработка на системата.....	148
7.2.1.1.	Методика за разработване на модела на данни.....	149
7.2.1.2.	Проектиране на базата данни	151
7.2.1.3.	Разработване на софтуерните модули (units)	153
7.2.1.4.	Процедура за управление на програмния код	153
7.2.1.5.	Инструменти за следене на прогреса на развитие на системата и разпределение на задачите	156
7.2.1.6.	Интегриране (build) на софтуерните модули	160
7.2.1.7.	Процедурата за управление на софтуерни грешки/несъответствия/проблеми	160
7.2.2.	Методика за внедряване	162
7.2.2.1.	Начин на прилагане на предлагания подход за внедряване на компонентите	164
7.2.3.	Методология за тестване.....	168
7.2.3.1.	Тестов план.....	169
7.2.4.	Методика за изпълнение на дейностите по обучение.....	179
	Примерна учебна програма	181
7.3.	Описание на архитектурата и подхода за реализация на предлаганото софтуерно решение.....	187
7.3.1.1.	Архитектура	187
7.3.1.2.	Подход за изграждане на надградената версия на системата	193
7.3.1.3.	Защита на информацията при изпълнение на проекта	210
7.3.1.4.	Подход към реализацията на изискванията за информационна сигурност	210

7.3.1.5.	Обосновка на избрания подход за реализация на изискванията за информационна сигурност	211
7.3.1.6.	Управление, идентифициране и предоставяне на права на потребители	212
7.3.1.7.	Идентификация и предоставяне на права.....	213
7.3.1.8.	Подход за реализация на ETL процеса за миграция на данни	213
7.3.2.	Функционални възможности на основните компоненти на реализацията	218
7.3.2.1.	Компонент „Събиране на данни за външнотърговски потоци“	219
7.3.2.2.	Компонент „Редактиране и обработка на микроданни“	219
7.3.2.3.	Компонент „Актуализиране на данни за търговците от СОРТ“	220
7.3.2.4.	Компонент „Предоставяне на данни на заинтересовани потребители“	220
7.3.2.5.	Компонент „Поддържане на общосистемна информация“	220
7.3.3.	Добавена нова и променена функционалност в надградената версия на системата.....	221
7.3.4.	Архивиране и проследяване на промените	224
7.4.	Изпълнение на специфичните изисквания към дейностите	227
7.4.1.	Дейност 1 Изготвяне на системен проект	227
7.4.1.1.	Описание на дейността	227
7.4.1.2.	Изпълнение на дейността „Изготвяне на системен проект“	227
7.4.1.3.	Резултати.....	227
7.4.2.	Дейност 2 Развитие на информационна система „Външна търговия“ за миграция на платформата на модул „Микроданни“ чрез използването на съвременни високотехнологични ИТ решения.....	228
7.4.2.1.	Описание на дейността	228
7.4.2.2.	Изпълнение на дейността.....	228
7.4.2.3.	Резултати.....	229
7.4.3.	Дейност 3 Развитие на информационна система „Външна търговия“ за добавяне на нови функционалности и промяна на съществуващите.	229
7.4.3.1.	Описание на дейността	229
7.4.3.2.	Изпълнение на дейността.....	229
7.4.3.3.	Резултати.....	232
7.4.4.	Дейност 4 Тестване на системата	232
7.4.4.1.	Описание на дейността	232
7.4.4.2.	Изисквания към изпълнение на дейността	232
7.4.4.3.	Резултати.....	232
7.4.5.	Дейност 5 Обучение.....	233

7.4.5.1.	Описание на дейността.....	233
7.4.5.2.	Изпълнение на дейността.....	233
7.4.5.3.	Резултати.....	233
7.4.6.	Дейност 6 Внедряване.....	233
7.4.6.1.	Описание на дейността.....	233
7.4.6.2.	Изпълнение на дейността.....	234
7.4.6.3.	Резултати.....	235
7.4.7.	Дейност 7 Гаранционно поддържане.....	236
7.4.7.1.	Описание на дейността.....	236
7.4.7.2.	Изисквания към изпълнение на дейността	236
7.4.7.3.	Резултати.....	237
8.	ДОКУМЕНТАЦИЯ.....	238
8.1.	Изисквания към документацията	238
8.1.1.	Методология за управление на проектната документация	239
8.2.	Прозрачност и отчетност	240
8.3.	Системен проект	241
8.4.	Техническа документация	241
8.5.	Протоколи	242
8.6.	Комуникация и доклади.....	242
8.6.1.	Встъпителен доклад	242
8.6.2.	Междинни доклади	242
8.6.3.	Окончателен доклад	243
9.	РЕЗУЛТАТИ	243

Списък на таблиците в документа:

Таблица 1 - Недостатъците на waterfall модела и преимуществата на RUP подхода	23
Таблица 2 - Таблица на съответствие на фазите и дейностите от проекта по Техническо задание (адаптиран RUP) с фазите/итерациите по РУП.....	26
Таблица 3 - Методи за изпълнение в предлаганата организация и методология	29
Таблица 4 - Роли и отговорности на членовете на екипа на Сمارт Системс	44
Таблица 5. Мерки за мониторинг и контрол във фазите на проекта.....	63
Таблица 6 - Оценка на Рисковата експозиция - матрица на вероятността	83
Таблица 7 – Рискова експозиция и степен на критичност	88
Таблица 8 - Работна версия на Регистъра на рисковете	89
Таблица 9 Приоритизация на рисковете и мерки за противодействие	94
Таблица 10 - Скала за степен на критичност	97

Таблица 7 – Примерен учебен план.....	181
Таблица 8 – Примерна учебна програма.....	181

Списък на фигурите в документа:

Фигура 1 Архитектура на RUP.....	21
Фигура 2 Гарантиране и повишаване качеството чрез предлаганите методи за изпълнение	32
Фигура 3 Организационната структура (органиграма) на Екипа за изпълнение на проекта при Сمارт Системс	51
Фигура 4 Канали за комуникация.....	55
Фигура 5 Модел на СУК, основана на процеси	56
Фигура 6 Корекция/добавяне на параметър	58
Фигура 7 Процес по управление на качеството	59
Фигура 8 Изводи за конкретен проект	65
Фигура 9 Контрол на качеството	69
Фигура 10 Гарантиране на качеството	71
Фигура 11 Модел на итерациите.....	73
Фигура 12 Подобрение на подходите за осигуряване и контрол на качеството.....	75
Фигура 13 Принципна схема на агрегатор на събития в система	105
Фигура 14 Модел на данни на общия интеграционен компонент.....	106
Фигура 15 Диаграма на итеративния подход и съответстващите му дейности и резултати	134
Фигура 16 Архитектурата на процесите за обектно ориентиран анализ и проектиране	141
Фигура 17 Диаграма на процесната архитектура за дизайн на компонентите.....	143
Фигура 18 Начало на диаграма за дейности	145
Фигура 19 Край на диаграма за дейности	145
Фигура 20 Действие в диаграма за дейности.....	146
Фигура 21 Преходи в диаграма за дейности.....	146
Фигура 22 Процес в диаграма за дейности	146
Фигура 23 Процес за проверка.....	147
Фигура 24 Обекти в диаграма за дейности	147
Фигура 25 Процедура за отстраняване на грешки/несъответствия/проблеми	161
Фигура 26 Процес на тестване	168
Фигура 27 Обобщени дейности по реализиране на тестовия процес	171
Фигура 28 Физическа архитектура на системата	188
Фигура 29 Логическа архитектура на предлаганото решение	189
Фигура 30 Концептуален модел за съхраняване на данни от изследвания	192
Фигура 31 Системна архитектура на реализацията на ИС – трислойна архитектура MVC.....	195
Фигура 32 Съхраняване на данни	197
Фигура 33 Многослойна Архитектура	199
Фигура 34 Централизирано съхранение и достъп до бизнес обектите	203

1. РЕЧНИК НА ТЕРМИНИ, ДЕФИНИЦИИ И СЪКРАЩЕНИЯ

1.1. Използвани съкращения

Акроним	Описание
АМ	Агенция „Митници“
АПК	Административно процесуален кодекс
БД	База данни
БД „MDE“	База данни “Micro-data exchange” (Обмен на микроданни)
БНБ	Българска народна банка
ВОД	Вътрешно-общински доставки (изпращане на стока или услуга от нас към друг стопански субект в ЕС)
ВОП	Вътрешно-общностни придобивания (пристигане при нас на стока, изпратена ни от друг стопански субект от ЕС)
БУЛСТАТ	Регистър Булстат
ДАЕУ	Държавна агенция „Електронно управление“
ДДС	Данък върху добавената стойност
ЕАД	Единен административен документ
ЕИО	Европейска икономическа общност
ЕС	Европейски съюз
ЗДОИ	Закон за достъп до обществената информация
ЗЕУ	Закон за електронно управление
ИС	Информационна система
ИСБС	Информационна система „Бизнес статистика“
ИСВТ	Информационна система „Външна търговия“
ИТ	Информационни технологии
КПИД	Класификация на продуктите по икономически дейности
МЗХГ	Министерство на земеделието, храните и горите
МИ	Министерство на икономиката
НАП	Националната агенция за приходите
НСИ	Национален статистически институт
ООН	Организация на обединените нации
РСЕ	Регистър на статистическите единици
СОРТ	Статистически общ регистър на търговците
СУБД	Система за управление на бази данни
СУСИ	Система за управление на сигурността на информацията
ТР	Търговски регистър
ТСБ	Териториално статистическо бюро

ДХЧО	Държавен хибриден частен облак
API	Application programming interface / Приложно програмен интерфейс
BEC	Board economic categories
NACE	Classification of economic activities (Nomenclature statistique des activités économiques dans la Communauté européenne")
SDK	Software development kit
SIMSTAT	Single Market Statistics
SITC	Standard International Trade Classification

1.2. Технологични дефиниции

Термин	Описание
Виртуална комуникацион-на инфраструктура	Инфраструктура, която на база съществуваща физическа свързаност, предоставена от ДАЕУ, предоставя възможност за изграждане на отделни и защитени виртуални мрежи за всяка една от структурите в сектора, при гарантиране на сигурен и защитен обмен на информация в тях.
Държавен хибриден частен облак	Централизирана на ниво държава информационна инфраструктура (сървъри, средства за съхранение на информация, комуникационно оборудване, съпътстващо оборудване, разпределени в няколко локации в помещения, отговарящи на критериите за изграждане на защитени центрове за данни), която предоставя физически и виртуални ресурси за ползване и администриране от секторите и структурите, които имат достъп до тях в зависимост от нуждите им, при гарантиране на високо ниво на сигурност, надеждност, изолация на отделните ползватели и невъзможност от намеса в работоспособността на информационните им системи или неоторизиран достъп до информационните им ресурси. Изолацията на ресурсите и мрежите на отделните секторни ползватели (е-Общини, е-Правосъдие, е-Здравеопазване, е-Полиция) се гарантира с подходящи мерки на логическо ниво (формиране на отделни клъстери, виртуални информационни центрове и мрежи) и на физическо ниво (клетки и шкафове с контрол на достъпа).
Софтуер с отворен код	Компютърна програма, която се разпространява при условия, които осигуряват безплатен достъп до програмния код и позволяват: Използването на програмата и производните на нея компютърни програми, без ограничения в целта; Промени в програмния код и адаптирането на компютърната програма за нуждите на нейните ползватели; Разпространението на производните компютърни програми при същите условия. Списък на стандартни лицензионни споразумения, които предоставят тези възможности, който може да бъде намерен в подзаконовата нормативна уредба към Закона за електронно управление или на: http://opensource.org/licenses .
Машинно четим формат	Формат на данни, който е структуриран по начин, по който, без да се преобразува в друг формат позволява софтуерни приложения да идентифицират, разпознават и извличат специфични данни, включително отделни факти и тяхната вътрешна структура.
Отворен формат	Означава формат на данни, който не налага употребата на специфична платформа или специфичен софтуер за повторната употреба на съдържанието и е предоставен на обществеността без ограничения, които биха възпрепятствали повторното използване на информация.

Метаданни	Данни, описващи структурата на информацията, предмет на повторно използване.
Официален отворен стандарт	Стандарт, който е установен в писмена форма и описва спецификациите за изискванията как да се осигури софтуерна оперативна съвместимост.
Система за контрол на версиите	<p>Технология, с която се създава специално място, наречено “хранилище”, където е възможно да се следят и описват промените по дадено съдържание (текст, програмен код, двоични файлове). Една система за контрол на версиите трябва да може:</p> <p>Да съхранява пълна история - кой, какво и кога е променил по съдържанието в хранилището, както и защо се прави промяната;</p> <p>Да позволява преглеждане разликите между всеки две съхранени версии в хранилището;</p> <p>Да позволява при необходимост съдържанието в хранилището да може да се върне към предишна съхранена версия;</p> <p>Да позволява наличието на множество копия на хранилището и синхронизация между тях.</p> <p>Цялата информация, налична в системата за контрол на версиите за главното копие на хранилището, прието за оригинален и централен източник на съдържанието, трябва да може да бъде достъпна публично, онлайн, в реално време.</p>
Първичен регистър	Регистър, който се поддържа от първичен администратор на данни - административен орган, който по силата на закон събира или създава данни за субекти (граждани или организации) или за обекти (движими и недвижими) за първи път и изменя или заличава тези данни. Например Търговският регистър е първичен регистър за юридическите лица със стопанска цел, Имотният регистър е първичен регистър за недвижима собственост.

1.3. Дефиниции за нива на електронизация на услугите

Термин	Описание
Ниво 1	Информация - предоставяне на информация за административни услуги по електронен път, включително за начини и места за заявяване на услугите, срокове и такси.
Ниво 2	Едностранна комуникация - информация съгласно дефиницията за Ниво 1 и осигурен публичен онлайн достъп до шаблони на електронни формуляри.
Ниво 3	Двустранна комуникация - заявяване и получаване на услуги изцяло по електронен път, включително електронно подаване на данни и документи, електронна обработка на формуляри и електронна персонална идентификация на потребителите.
Ниво 4	Извършване на сделки или транзакции по услуги от Ниво 3, включващи онлайн разплащане или доставка.

2. ПРЕДСТАВЯНЕ НА УЧАСТНИКА - СМАРТ СИСТЕМС 2010 ЕООД

От своето създаване „Смарт системс 2010“ ЕООД е фокусирана като дейност в областта на разработката, внедряването и поддръжката на информационни системи в българската съдебната система. Компанията се специализира в разработката на софтуерни решения и предоставянето на консултации в сферата на оптимизацията и реструктуриране на бизнес процеси, управление на проекти, както и внедряването и интеграцията на информационни системи.

Целта на „Смарт системс 2010“ ЕООД е да бъде катализатор за усъвършенстване на организацията на работа и управление в българските съдилища, чрез все по-широко прилагане на информационните технологии в автоматизацията на работните процеси в съдебната система.

Екипът на „Смарт системс 2010“ ЕООД е разработил, внедрил и поддържа и разширява функционалността на Автоматизираната система за управление на делата, внедрена в над 20 съдилища, включително Върховния касационен съд, Софийски апелативен съд, Административен съд София – град, Софийски районен съд и ред други окръжни, районни и административни съдилища. АСУД успешно обслужва работата в повечето от най-натоварените съдилища в България, като опитът от оптимизациите извършени в тези съдилища дава увереност и ясна визия за реализацията на единна и централизирана система за управление на дела. В рамките на развитието на АСУД, „Смарт системс 2010“ ЕООД участва успешно в редица ключови проекти свързани с интеграцията и разширяването на обхвата на системите за управление на дела:

- "Разработване и внедряване на единен портал на електронното правосъдие, и обучение за работа с единния портал", по проект: „Електронно правосъдие – проучване и изграждане на единна комуникационна и информационна инфраструктура и единен електронен портал на съдебната власт” (ЕПЕП)
- „Разширяване и адаптация на функционалността на модула за разпределение на дела в Автоматизирана система за управление на дела (АСУД), разработена и одобрена от ВСС, по проект на Американската агенция за международно развитие (ААМР), с цел неговата централизация, отговаряща на изискванията за функционалност и сигурност към системите за разпределение на съдебни дела, одобрени с решение на ВСС по протокол №62/18.12.2014г.“ (ЦСРД)
- Изпълнение на дейности по реализация и внедряване на информационна система за управление на информацията за натовареността на съдилищата (СИНС)
- Обособена позиция № 2: “Разработка, внедряване и интеграция на приложен софтуер и доставка на базов софтуер” от проект „Доставка и инсталиране на хардуер и базов софтуер, разработка, интеграция и внедряване на приложен софтуер, необходими за модернизиране на съществуващите системи на управление на делата, с цел електронно съобщаване и призоваване“ (Система за електронно призоваване)

Екипът на фирмата има експертиза в следните области:

- Бизнес консултиране (управление и оптимизация на бизнес процеси, бизнес моделиране и анализ, спецификация на изискванията);
- Проектиране и разработка на информационни системи;

- Управление на проекти;
- Интеграция на приложения;
- Системна интеграция

Използваните технологии в „Смарт системс 2010“ ЕООД са:

- Microsoft – ASP.NET Framework, MS SQL Server;
- IBM Lotus Domino;

Консултантската дейност дава възможност на „Смарт системс 2010“ ЕООД да идентифицира и разбере по-добре бизнес нуждите на своите клиенти и областите, които се нуждаят от подобрене. Чрез разработката на софтуерни приложения се предоставят технически решения, осигуряващи реални ползи за бизнеса.

3. ЗА ПРОЕКТА

3.1. Общи и специфични цели на проекта

Проектът е насочен към поддържане на ефективно функциониране при непрекъснато нарастващия обем данни, осигуряване на информационната сигурност, бързодействието, надеждността и актуалността на съществуващата ИСВТ, както и към задоволяване на нарастващите нужди на основните потребители. Това може да бъде постигнато в резултат от развитието на системата чрез използване на съвременни ИТ решения, надграждане на наличните и добавяне на нови функционалности, като се вземат предвид настъпилите и предстоящите промени в Европейското законодателство и изискванията на Европейската комисия.

Постигането на общата цел ще бъде реализирано чрез следните специфични цели, съответстващи на планираните по проекта дейности:

- Специфична цел 1: Миграция на платформата на модул „Микроданни“ на ИСВТ чрез използването на съвременни високотехнологични ИТ решения;
- Специфична цел 2: Добавяне на нови функционалности и промяна на съществуващите функционалности на ИСВТ;
- Специфична цел 3: Интегриране на ИСВТ с други ИС на НСИ – СОРТ, РСЕ и ИСБС. ИСВТ използва информация от други ИС в НСИ, за тази цел следва да се изградят функционалности за автоматичен обмен на данни по зададени критерии;
- Специфична цел 4: Гаранционна поддръжка на ИСВТ за срок от 36 месеца, която включва:
 - Предоставяне на консултации на ВЪЗЛОЖИТЕЛЯ относно администриране, настройване, конфигуриране, мониторинг и експлоатация на системата;
 - Отстраняване на възникнали проблеми в приложния софтуер на системата;
 - Отстраняване на възникнали проблеми в инсталациите, конфигурациите и настройките на сървърите на системата;

- Реализиране на малки промени и/или допълнения в приложния софтуер, конфигурацията и настройките на системата и сървърите, на които е инсталирана.

Постигането на целите на настоящото ТЗ ще осигури възможност за работа със системата чрез новите версии на най-популярните Интернет- браузъри (в това число Microsoft Edge, Microsoft Internet Explorer, Google Chrome, Mozilla Firefox).

Постигането на целите на проекта ще улесни работата в областта на статистиката на външната търговия със стоки и ще подобри като цяло качеството на външнотърговските данни, по-конкретно чрез:

- Осигуряване на бързодействие и надеждност на Информационна система „Външна търговия“ при непрекъснато нарастващ обем данни и повишени изисквания за информационната сигурност;
- Намаляване на времето и необходимите ресурси от страна на НСИ за производство на статистически данни за външната търговия със стоки за осигуряване на необходимите данни за ключовите потребители, включително Евростат и държавите членки;
- По-пълно използване на административни данни като източник на информация за изчисляване на допълнителни показатели за външната търговия със стоки, включително за вноса и износа на стоки по бизнес характеристики на предприятията;
- Използване на добрите европейски практики в областта на статистиката за международната търговия със стоки;
- Създаване на възможност за свързване на микроданните между различни изследвания, анализ на резултатите и в следствие на това, подобряване на качеството на информацията от статистиката за външната търговия със стоки;

3.2. Обхват на обществената поръчка

Описаните в т. 3.1 цели се осъществяват с изпълнението на следните основни дейности, които формират обхвата на проекта:

- Дейност 1 - Разработване на системен проект ;
- Дейност 2 - Развитие на информационна система „Външна търговия“ за миграция на платформата на модул „Микроданни“ чрез използването на съвременни високотехнологични ИТ решения;
- Дейност 3 - Развитие на информационна система „Външна търговия“ за добавяне на нови функционалности и промяна на съществуващите;
- Дейност 4 - Тестване на системата;
- Дейност 5 - Провеждане на обучение на служителите на НСИ за работа с ИСВТ;
- Дейност 6 – Внедряване;
- Дейност 7 - Гаранционна поддръжка.

3.3. Целеви групи

Целевите групи, към които е насочен проектът, обхващат:

- Национален статистически институт;
- Ключови потребители – БНБ, Евростат, държавите членки, ООН, министерства;
- Други потребители - правителства, предприятия, академични и европейски изследователи, държавни администрации, браншови и международни организации, бизнес среди, физически лица и други лица, осъществяващи публични функции.

3.4. Очаквани резултати

Очакваните резултати от изпълнението на настоящата поръчка са:

- Внедрена в експлоатация ИСВТ, базирана на съвременни ИТ решения, с надградени и добавени нови функционалности, с бързодействие и сигурност, необходими за ефективното производство и съхранение на статистическите данни за износа и вноса на стоки;
- Внедрена в експлоатация ИСВТ, използваща съвременни средствата за изготвяне на навременна и висококачествена статистика за износа и вноса на стоки, отговаряща на нарасналите нужди на основните потребители;
- Разработена документация за ИСВТ – техническо описание, ръководство за потребители, ръководство за администриране и поддръжка;
- Проведено обучение за работа с усъвършенстваната ИСВТ на служителите на НСИ.

3.5. Период на изпълнение

Периодът на изпълнение е 12 (дванадесет) месеца, считано от датата на подписване на договора, но не по-късно от 31.12.2020 г..

Изготвен е подробен график, в който са конкретизирани сроковете за изпълнение на всяка дейност и поддейност от настоящата поръчка. Графикът за изпълнение е съобразен с продължителността на дейността и не надвишава 12 месеца от дата на сключване на договора.

4. Организация и методология за управление на проекта

При изпълнението на поръчката, Сمارт Системс т ще спазва всички нормативни изисквания по отношение на дейността на НСИ и електронното управление в Република България.

4.1. Общи организационни принципи

При изпълнение на настоящата поръчка ще се спазват всички нормативни изисквания по отношение на дейността на Националния статистически институт и електронното управление в Република България.

За да може да бъде гарантирано постигане на желаните резултати от проекта, така че да се покрие пълният набор от компетенции и ноу-хау, необходими за изпълнение на предмета на поръчката, а също така да се гарантира и достатъчно ниво на ангажираност с изпълнението и проблемите на проекта ще бъдат спазени и утвърдените хоризонтални и вертикални принципи на организация на изпълнението на предмета на обществената поръчка.

Хоризонтален принцип – ще се ангажират специалисти от различни звена, така че да се покрие пълният набор от компетенции и ноу-хау по предмета на проекта и същевременно екипът да усвои новите разработки на достатъчно ранен етап, така че да е в състояние пълноценно да ги използва и развива и след приключване на проекта;

Вертикален принцип – ще се осигури участие на експерти и представители на различните управленски нива, така че управленският екип да покрива както експертните области, необходими за правилното и качествено изпълнение на проекта, така и управленски и организационни умения и възможности за осъществяване на политиката във връзка с изпълнението на проекта. Чрез участие на ръководители на звената – ползватели на резултата от проекта, ще се гарантира достатъчно ниво на ангажираност на институцията с проблемите на проекта.

Дейностите по управление на изпълнението ще включват като минимум управление на реализацията на всички дейности, посочени в настоящата обществена поръчка и постигане на очакваните резултати, както и разпределението на предложените участници в екипа за управление на поръчката по роли, график и дейности при изпълнение на настоящата обществена поръчка.

Създаването на правилна организация за изпълнение на даден проект е един от най-критичните фактори за успеха на проекта.

Управлението на настоящия проект на проекта ще осигури:

- координиране на усилията на експертите от страна на Сمارт Системс и Възложителя и осигуряване на висока степен на взаимодействие между членовете на проектния екип;
- оптимално използване на ресурсите;
- текущ контрол по изпълнението на проектните дейности;
- разпространяване навреме на необходимата информация до всички участници в проекта;
- идентифициране на промени и осигуряване на техните анализ и координация;
- осигуряване на качеството и полагане на усилия за непрекъснато подобряване на работата за удовлетворяване на изискванията на участниците в проекта.

Методологията, която ще бъде използвана за Управление на проекта ще включва подробно описание на:

- фазите на проекта;

- организация на изпълнение:
 - структура на екипа на Сمارт Системс ;
 - начин на взаимодействие между членовете на екипа на Смарт Системс ;
 - връзки за взаимодействие с екипа на Възложителя;
- проектна документация:
 - видове доклади;
 - техническа и експлоатационна документация;
 - време на предаване;
 - съдържание на документите;
 - управление на версиите;
- управление на качеството;
- график за изпълнение на проекта.

В графика са описани дейностите и стъпките за тяхното изпълнение максимално детайлно, като е показана логическата връзка между тях. В графика са посочени датите за предаване на всеки от документите, изготвени в изпълнение на обществената поръчка.

Подробно описание на Методологията за изпълнение и управление на проекта се намира в т. 4.2

4.2. Методология за изпълнение на проекта

Предлагаме Методологията за управление на процеса по софтуерна разработка **RUP /Rational Unified Process/**, да бъде използвана като водеща методологична рамка за изпълнение на поръчката.

Предложената методология RUP /Rational Unified Process/ съответства с най-добрите световни практики и препоръки.

RUP е итеративен и инкрементален процес, който позволява непрекъсната еволюция и подобрене на информационно-технологичните системи в множество планирани и управлявани фази. Базиран е на следните **шест най-добри практики**:

- **Итеративна разработка на софтуера** - позволява по - добро разбиране на изискванията и постепенното им надграждане, намалява рисковете свързани с обхвата и времевия график;
- **Управление на изискванията** – позволява тяхното правилно описание, проследяване и документиране;
- **Използване на компонентно ориентирана архитектура** – позволява изграждането на гъвкава и преизползваема архитектура;
- **Визуално моделиране** – позволява комуникиране на различни аспекти на софтуера между основните участници;
- **Проверка и управление на качеството** – позволява системна проверка и контрол на аспектите на системата, свързани с нейната надеждност и производителност;
- **Управление и контрол на промените** – позволява мониторинг, контрол и

проследяване на промените за осъществяване на итеративния процес.

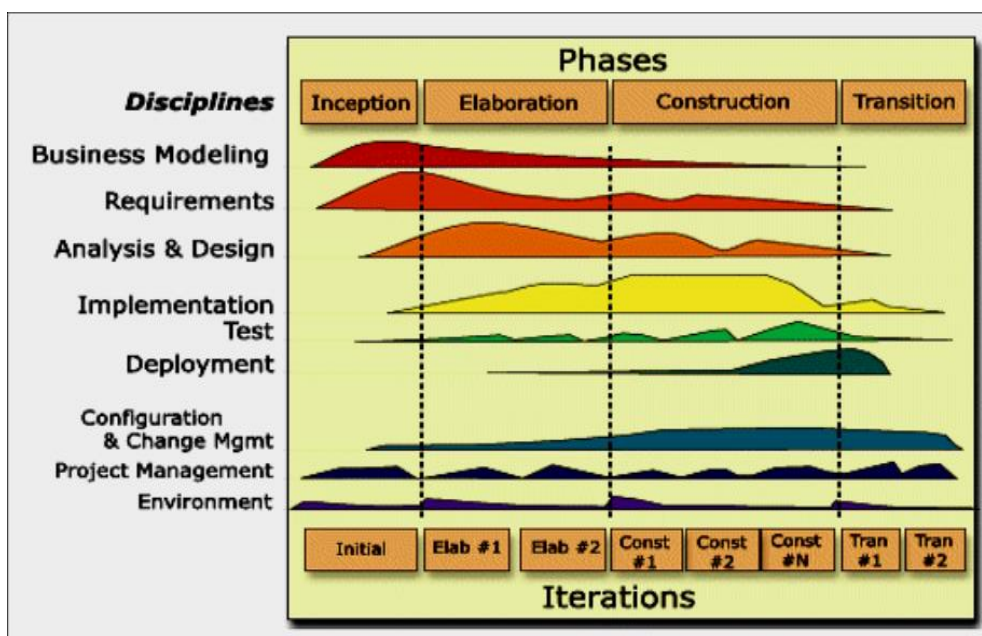
Софтуерният цикъл на унифицирания процес е разделен във времето на **четири последователни фази (Планиране, Детайлизиране, Изграждане и Предаване)**. Всяка фаза е разделена на итерации, една или няколко. Итерациите могат да бъдат разглеждани като под проекти с ясна последователност от дейности с определен план и оценителни критерии, завършващи с артефакт за бенефициента.

По време на итерация се изпълняват дейности, отнасящи се до идентифициране и анализиране на изискванията, имплементация, тестване.

Множество от тестови сценарии ще се конструират с помощта на методите на **Unified Modelling Language**. За изпълнението на Sequence Diagrams и Activity Diagrams ще бъдат използвани изпитани добри практики.

Артефактите в началните итерации са главно документи, характеризиращи изискванията и съдържащи аналитични и UML модели. Последващите итерации произвеждат софтуерни версии, в които има имплементирана желаната системна функционалност. Завършващите итерации се съсредоточават върху тестването, поправяне на грешки, и евентуално развитие на системата. Всяка итерация разширява знанията на проектния екип и прибавя нова функционалност инкрементално. Тази техника позволява динамична реакция на променящите се изисквания. На всяка итерация особено се акцентира на управлението и на оценката до каква степен за изпълнени набелязаните цели. Това намалява до минимум възможността от риска за неизпълнение на проекта.

RUP е ориентиран към софтуерния инженерен процес. Предлага подход за разделяне на задачите и отговорностите по дисциплини. Неговата цел е да подsigури разработка на софтуер с високо качество, който удовлетворява нуждите на крайните потребители в рамките на предварително определен бюджет и срокове.



Фигура 1 Архитектура на RUP

Фигурата по-горе илюстрира **архитектурата на RUP**, която има две измерения:

- **Хоризонталната ос** представлява времето и показва цикъла на живот на процеса. Това първо измерение илюстрира динамичността и е изразена с фази, итерации и събития;
- **Вертикалната ос** представлява дисциплините (работните потоци), които групират логически дейностите. Второто измерение илюстрира статичността на процеса изразена с компоненти, дисциплини, дейности, артефакти, роли.

4.2.1. Обосновка на използването на RUP за успешното изпълнение на проекта

- В световен мащаб RUP е една от най-добрите методологии и инструментариуми за прилагането на процесния подход при реализацията на мащабни и комплексни ИТ проекти, какъвто е и настоящият;
- Множество успешно реализирани и функциониращи в момента информационни системи и ИТ проекти са разработени и внедрени при използването на RUP, налагайки я като водещата методологична рамка, доказала приложимостта и предимствата си за успешното реализиране на мащабни и сложни проекти.
- Използването на RUP като водеща методологична рамка при успешното разработване на предишни проекти е довело до натрупване на ценен опит и знания по прилагането на тази методология от експертите на „Смарт Системс 2010“ ЕООД.
- Определените от Възложителя в т.б. от Техническото задание минимални етапи на проекта по същество представляват фазите и итерациите на RUP, адаптирани към организационната среда и спецификите на конкретния проект.

Въпреки факта, че в последните десетилетия обектно ориентираното програмиране се утвърждава като доминираща парадигма в разработката на софтуер, съществуват диаметрално противоположни методологии за разработка, които могат да осигурят прилагането на принципите на обектно ориентираното програмиране в даден проект– от така наречените „леки“ и „динамични“ гъвкави методологии залагащи на по-малко структуриране и формализиране (Agile Methodologies – <http://agilemanifesto.org/>) от типа на Scrum, Extreme Programming (XP), през методологии като RUP и Dynamic Systems Development Method (DSDM), които позволяват комбиниране на предимствата на различни методологии, до така нар. „тежки“ методологии като традиционния каскаден (waterfall) модел.

Използването на „леки“ методологии с минимално ниво на формализиране в голям проект като настоящият не би било удачен избор, поради големия обхват на реализацията, големия екип за изпълнение, фиксираната продължителност на проекта и невъзможността за промяна на сроковете за изпълнение по него.

Традиционният (waterfall) подход за софтуерна разработка се отличава с това, че дейностите протичат в процес, при който фазите на разработка на софтуера следват точно определен ред - линейно и последователно. Това означава, че всеки от етапите в процеса на разработка започва, само когато предишната фаза е напълно завършена. При стриктно спазване на методологията връщане към предишна фаза за преправяне на продукта поради промяна на изискванията, не се допуска.

Изискването на техническото задание за разработка с етапи на анализ, проектиране, разработване, тестване и внедряване прави възможно прилагането на тежък подход като waterfall, но от гледна точка на относително краткия период за реализация на проекта, използването на по-гъвкавия подход е напълно подходящо за описания в организацията на изпълнението tailored RUP. При този итеративен подход ще се извършат чести представяния на резултата от разработката, така че Възложителя ще има поглед върху напредъка и не е нужно да се чака до края на проекта, за да види резултата. При този подход Възложителя ще участва пряко в отделните етапи на проекта, като работата на разработчиците може да бъде обсъждана по време на итерациите на разработка. Това спомага за минимизиране на недоразуменията, по-ясно разпознаване на нуждите на клиента и адаптиране на софтуера към променящите се изисквания. При RUP преправяне на продукта поради промяна на изискванията се допуска и се реализира по време на следващата итерация (цикъл) от софтуерната разработка.

Следната сравнителна таблица показва ясно разликите в двата подхода - недостатъците на waterfall модела и преимуществата на RUP подхода, които допринасят за качествено изпълнение.

Таблица 1 - Недостатъците на waterfall модела и преимуществата на RUP подхода

Слаби страни на традиционния „waterfall” технологичен подход	Преимущества на RUP подхода с оглед постигане на качествено изпълнение
Всички изисквания към разработките (софтуера) трябва да са предварително ясни и дефинирани в детайли преди да стартира софтуерната разработка.	Изискванията към разработките (софтуера) се дефинират по време на етап Бизнес и системен анализ, когато Сمارт Системс т е събрал достатъчно информация от Възложителя и обхвата на поръчката е окончателно дефиниран. Дефинирането на изискванията по време на самото изпълнение, а не преди неговият старт (както е при waterfall подхода) е предпоставка за по-точно и коректно извличане и формулиране на изискванията и е предпоставка за качествено изпълнение.
Скъп и бавен процес	RUP е олекотен и по-бърз за изпълнение подход, защото позволява няколко екипа да работят едновременно. Интерактивността на подхода, която се състои в постъпково (поетапно) надграждане на продукта позволява вкарването на промени в изискванията дори и по време на разработката без това да оскъпява допълнително изпълнението.
Подходът не позволява да се вкарват промени в процеса на разработка, защото това излиза много скъпо и изисква допълнително (непредвидено) време	RUP подходът позволява да се вкарват промени в процеса на разработка, за разлика от waterfall подхода. При waterfall подхода промените са скъпи, защото обикновено тяхната нужда се вижда след като продукта е завършен и реализирането на промяната изисква

Слаби страни на традиционния „waterfall” технологичен подход	Преимущества на RUP подхода с оглед постигане на качествено изпълнение
	допълнително време и то в края на проекта, както и големи промени на много места в кода.
Крайният потребител няма много възможности за участие в проекта по отношение на изискванията към разработките, защото те са предварително дефинирани	В RUP подхода експерти на Възложителя (потребители на софтуера) имат принос във формулирането на изискванията по време на етап Бизнес и системен анализ, когато се провеждат работните срещи между екипите на двете страни с оглед запознаване с бизнес процесите и извличане на изискванията към софтуерните разработки.
Много често проблемите се установяват едва в етапа на тестване на софтуера, когато продукта е завършен	При RUP в края на всяка итерация (фаза от разработката) Смарт Системс т представя за преглед и одобрение на Възложителя някаква готова част от софтуера, която, ако не отговаря на изискванията и/или има грешки, се променя и подобрява по време на следващата итерация по време на етап Разработка.
Подходът се характеризира с обемна техническа документация, която в много случаи е неразбираема за крайния потребител	При RUP документацията е по-малка, но за сметка на това ориентирана към крайния потребител.

Постигането на конкретните резултати на проекта и техническата реализация ще се осъществи чрез подхода за софтуерна разработка RUP, който ще осигури реализацията на техническите разработки като гарантира, че Възложителят и „Смарт Системс 2010“ ЕООД ще могат да контролират заедно развитието на проекта.

Изборът на RUP се базира на възможностите, които този гъвкав и олекотен подход предоставя - за управление и контролиране на софтуерната разработка поради разбиването ѝ на итерации, всяка от които завършва с готова/разработена част софтуер, която може да се прегледа и тества от Възложителя на ранен етап (не в края на цялостната разработка) и в случай на необходимост да се внесат корекции от разработчиците на „Смарт Системс 2010“ ЕООД преди крайния срок за реализиране на софтуерното приложение. Накратко, RUP подходът позволява ранна валидация на готови „части“ софтуер, получаването на обратна връзка от Възложителя по време на софтуерната разработка, а не след нейното приключване и ранно внасяне на промени, корекции и допълнения в кода.

Освен това RUP подходът ще гарантира доброто управление на проекта, което както е посочено в Техническото задание, ще осигури:

- координиране на усилията на експертите от страна на Смарт Системс и Възложителя и осигуряване на висока степен на взаимодействие между членовете на проектния екип;
- оптимално използване на ресурсите;
- текущ контрол по изпълнението на дейностите;

- разпространяване навреме на необходимата информация до всички участници в проекта;
- идентифициране на промени и осигуряване на техните анализ и координация;
- осигуряване на качеството и полагане на усилия за непрекъснато подобряване на работата за удовлетворяване на изискванията на участниците.

4.2.2. Фази на проекта

Фазите при изпълнение на проекта ще бъдат следните:

- Анализ на данните и изискванията
- Изготвяне на системен проект
- Разработване на софтуерното решение
- Тестване
- Внедряване
- Обучение
- Гаранционна поддръжка

Подробно описание на подхода за изпълнение на дейностите във фазите по проектиране може да бъде намерено в т. 7.1. Предложение за извършване на дейностите по анализ и проектиране от Техническото предложение.

4.2.3. Функционални особености на приложения подход – съотнесимост на методологията RUP към фазите на конкретния проект

Както беше изложено в началото, една от причините да предложим използването на RUP като водеща методологична рамка за изпълнение на поръчката е, че определените от Възложителя в Техническото задание дейности и етапи на проекта по същество съответстват на фазите и дейностите в итерациите на RUP, адаптирани към организационната среда и спецификите на конкретния проект.

Съчетанието от дефинираните от Възложителя етапи на проекта, съответстващи по същество на RUP и краткия срок на проекта, който налага фазите по същество да се припокриват и да се търси максимално ранно запознаване на клиента в продукта на разработката, налагат RUP да бъде „адаптиран“ към организационната среда и спецификите на конкретния проект.

“Адаптирането“ (tailoring) е породен от реалния живот подход, припознат и допустим във всяка една от методологиите, използвани в ИТ сферата – RUP, SCRUM, PRINCE2, PMP и др.

Прилагайки подходът на „адаптиране“ се дава възможност за конкретния проект да се постигне съчетание от най-подходящите елементи от няколко методологии, като една от тях все пак остава водеща. Отчитайки съвкупно всички изисквания на Възложителя, предлагаме адаптирането на RUP към организационната среда и спецификите на настоящия проект да се състои в:

- прилагането на RUP като водеща методологична рамка по отношение на Организация за изпълнение поръчката;
- прилагането на RUP като водеща методологична рамка по отношение на Методологията за управлението на проекта, допълнен с цел да се отговори на посочените по-горе специфични изисквания с елементи от методологията SCRUM – прилагане на месечни итерации (спринтове) в етапи на проекта „Разработване на софтуерното решение“ и „Тестване“. Считаме, че прилагането на месечни итерации (спринтове) в другите етапи на проекта няма да е ефективно;
- прилагането на RUP като водеща методологична рамка по отношение на подхода за реализация на дейностите по проекта (изложена в т.4.2 от настоящия документ).
- Всяка от фазите на проекта намира свое естествено отражение в структурата на фазите от RUP, както е отразено в следващата таблица

Таблица 2 - Таблица на съответствие на фазите и дейностите от проекта по Техническо задание (адаптиран RUP) с фазите/итерациите по РУП

Етап на проекта по ТЗ (адаптиран RUP)	Фаза/ Итерация по RUP	Очакван резултат
Етап 1 Анализ на данните и изискванията	Детайлизиране	<p>Специфицирани потребителските изисквания към системата.</p> <p>Във фазата на проучване, по време на която ще се дефинират потребителските нужди, ще се проведат предварителни тестове с потребители и ще се изработи план, по който да се адресират идентифицираните нужди.</p> <p>Изготвянето на системния проект включва следните основни задачи:</p> <ul style="list-style-type: none"> • Определяне на концепция на информационната система на базата на техническото задание; • Дефиниране на детайлни изисквания и бизнес процеси, които трябва да се реализират в системата; • Дизайн на информационната система, хардуерната и комуникационната инфраструктура; • Изготвяне на план за техническа реализация; • Определяне на потребителския интерфейс.

<p>Етап 2 Изготвяне на системен проект</p> <p>Етап 3 Разработка на системата</p>	<p>Изграждане</p>	<p>Етапът на разработка включва изпълнението на следните задачи:</p> <ul style="list-style-type: none"> • Разработка на прототип, който трябва да бъде одобрен от Възложителя и въз основа на който трябва да се разработи цялата система; • Разработка на модулите на информационната система съгласно изискванията на настоящото техническо задание и системния проект; • Провеждане на вътрешни тестове на системата (в среда на разработчика); • Изготвяне на детайлни сценарии за провеждане на приемателните тестове за етапи „Тестване“ и „Внедряване“ на проекта.
<p>Етап 4 Тестване</p> <p>Етап 5 Внедряване</p> <p>Етап 6 Обучение</p>	<p>Предаване</p>	<ul style="list-style-type: none"> • Извършено тестване, в съответствие с Плана за тестване. • Проведени функционални тестове, интеграционни тестове, тестове за производителност на всички нови функционалности, а също и регресионни тестове и тестове на сигурността на бъдещата система. • Внедрени в продукционна среда софтуерни компоненти. • Изготвяне на документацията съпътстваща разработените софтуерни компоненти. • Внедряване на софтуерното решение в информационната и комуникационна среда на <i>НСИ</i>. Това включва инсталиране, конфигуриране и настройка на програмните компоненти на системата в условията на експлоатационната среда на <i>НСИ</i>. • Организиране и провеждане на обучение за потребителите на софтуерното решение.

Етап 7 Гаранционна поддръжка	Експлоатация	Гаранционна поддръжка на ново разработените функционалности за срок от 36 месеца, в съответствие с изискванията посочени в Техническото задание.
-------------------------------------	--------------	--

4.2.4. Използвани концепции, методи и инструменти

Методологията за управлението на проекта, при водещо прилагане на методологията RUP се основава на дисциплината Управление на проекта (Project Management) от тази методологична рамка.

Целта на дисциплината Управление на проекта от RUP е да се направляват и контролират дейностите, които се извършват по време на изпълнението на проекта/поръчката. Това най-общо включва управление на риска и качеството, управление на хората/екипа за изпълнение (раздаване на задачи, следене на прогреса) и координиране с хора и системи външни за проекта.

Критичните дейности/задачи в рамките на управлението на проекта по RUP са:

- стартиране на проекта;
- управление на екипа за изпълнение;
- непрекъснато подобряване на взаимодействието с екипите на Възложителя и с външни екипи и експерти;
- управление на риска и качеството;
- планиране на дейностите и техните изпълнители;
- управление на итерациите/фазите;
- приключване на итерации/фази и на целия проект.

За да бъде разбрана философията и логиката на Методиката за оперативно управление на проекта, базирана на RUP още веднъж ще акцентираме на това, че **RUP е итеративен и инкрементален процес**, който позволява **непрекъсната еволюция и подобрене на информационно - технологичните системи** в множество планирани и управлявани фази и е базиран е на следните шест най – добри практики:

- 1) Итеративна разработка на софтуера - позволява по-добро разбиране на изискванията и постепенното им надграждане, намалява рисковете свързани с обхвата и времевия график;
- 2) Управление на изискванията – позволява тяхното правилно описание, проследяване и документиране;
- 3) Използване на компонентно ориентирана архитектура – позволява изграждането на гъвкава и преизползваема архитектура;
- 4) Визуално моделиране – позволява комуникиране на различни аспекти на софтуера между основните участници;
- 5) Проверка и управление на качеството – позволява системна проверка и контрол на

аспектите на системата, свързани с нейната надеждност и производителност;

- б) Управление и контрол на промените – позволява мониторинг, контрол и проследяване на промените за осъществяване на итеративния процес.

Дейностите и задачите по управление на проекта ще бъдат съобразени с всички нормативни изисквания по отношение на дейността и електронното управление в Република България.

Използваните инструменти за управление на проекта ще бъдат системата за управление на задачите **Redmine**, която ще бъде използвана за управление на задачите и следене на напредъка на екипа по изпълнението, както и **MS Project**, за изготвяне и поддържане на подробния план на проекта.

4.2.5. Адаптиране на предлаганата методика към спецификата на проекта

Методите за изпълнение на предлаганата организация и методология за изпълнение на договора включват набор от техники и инструменти за осигуряване качествено изпълнение на дейностите в обхвата на договора.

Предвид спецификата на проекта и разнообразието от дейности с различен фокус, които в същото време са взаимосвързани и подчинени на обща цел ще бъдат използвани разнообразни методи за изпълнение на договора, включващи както технологични инструменти - софтуер за моделиране на процеси, софтуер за управление на проекти, софтуер за създаване на документи и управление на версии на документи и др., така и методични средства - техники за събиране на изисквания за разработка на софтуер, техники за провеждане на сравнителен анализ, функционален анализ и др. Изборът на технологичен инструмент или средство е обусловен от опита и експертизата „Смарт Системс 2010“ ЕООД за прилагане на съответния метод в други проекти.

В текущата точка се разглеждат само методите за изпълнение по отношение организацията и методологията за изпълнение на договора като цяло.

Методите за изпълнение в предлаганата организация и методология за изпълнение на договора включват:

Таблица 3 - Методи за изпълнение в предлаганата организация и методология

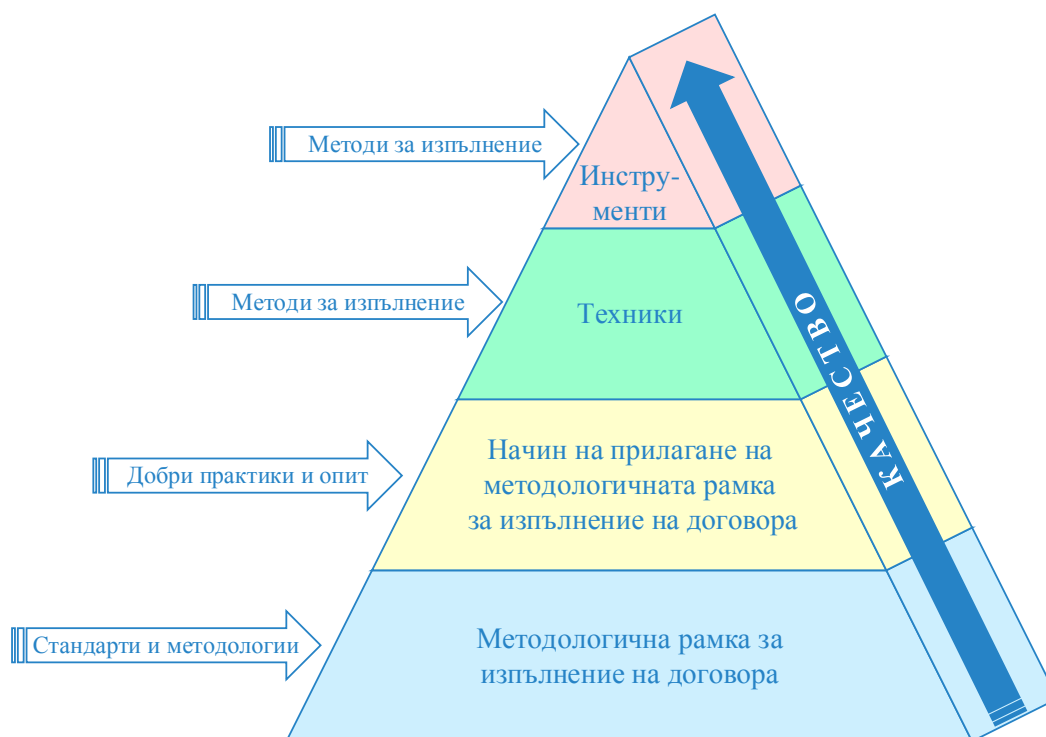
Фаза на изпълнение на договора	Методи за изпълнение Техники	Инструменти
1. Инициране и стартиране на договора	Провеждане на Встъпителна среща с Възложителя за очертаване рамката на договора по отношение на цели, очаквания, етапи, организация и комуникация	<ul style="list-style-type: none">• Телефон, електронна поща, работна среща;• Софтуер: MS Outlook; MS Word, MS Excel, MS Project

2. Планиране на изпълнението на договора	на на	Преглед и актуализиране документацията, изготвена към момента на кандидатстване за изпълнение на обществената поръчка	<ul style="list-style-type: none"> • Телефон, електронна поща, работна среща; • Софтуер: MS Outlook; MS Word, MS Excel, MS Project, MS Visio, MS Power Point
2.1. Изготвяне и приемане на план-график за изпълнение на дейностите по договора	на на	Актуализиране и съгласуване на план-график за изпълнение на дейностите по договора съгласно предложения подход.	<ul style="list-style-type: none"> • Телефон, електронна поща, работна среща; • Софтуер: MS Outlook; MS Word, MS Excel, MS Project
2.2. Създаване на необходимата организация на изпълнение на договора, формиране и определяне на структурата на екипа за изпълнение на договора	на на на на на на на на	Сформиране на екип съобразно предложената методология.	<ul style="list-style-type: none"> • Телефон, електронна поща, работна среща; • Софтуер: MS Outlook; MS Word, MS Excel, MS Project, MS Visio, MS Power Point
2.3. Управление на комуникациите	на на	Съгласуване на начин и връзки на взаимодействие между екипите на страните на договора, съгласно предложената методология.	<ul style="list-style-type: none"> • Телефон, електронна поща, работна среща; • Софтуер: MS Outlook; MS Word, MS Excel, MS Project, MS Visio, MS Power Point
2.4. Управление на качеството	на на	Съгласуване на подход за управление на качеството, съгласно описанието.	<ul style="list-style-type: none"> • Телефон, електронна поща, работна среща; • Софтуер: MS Outlook; MS Word, MS Excel, MS Project, MS Visio, MS Power Point
2.5. Управление на риска		Съгласуване на подход за управление на риска, съгласно описанието.	<ul style="list-style-type: none"> • Телефон, електронна поща, работна среща; • Софтуер: MS Outlook; MS Word, MS Excel, MS Project, MS Visio, MS Power Point
3. Изпълнение на дейностите по договора	на на	Методите за изпълнение на специфичните дейности в обхвата на проекта	<ul style="list-style-type: none"> • Телефон, електронна поща, работна среща; • Софтуер: MS Outlook; MS Word, MS Excel,

		MS Project, MS Visio, MS Power Point
4. Мониторинг и контрол на проектните дейности. Представяне на работата пред Възложителя	<ul style="list-style-type: none"> • Текущ мониторинг; • Вътрешен контрол; • Докладване на работата чрез: <ul style="list-style-type: none"> ○ Встъпителен доклад; ○ Междинни доклади; ○ Окончателен доклад; 	<ul style="list-style-type: none"> • Телефон, електронна поща, работна среща; • Софтуер: MS Outlook; MS Word, MS Excel, MS Project, MS Visio, MS Power Point
5. Поддръжка и гаранционно обслужване	Приемане и обслужване на заявки за проблеми	<ul style="list-style-type: none"> • Телефон, електронна поща, работна среща; • Софтуер: Система за регистриране и управление на заявки за проблеми и дефекти; MS Outlook; MS Word, MS Excel, MS Project
6. Административно приключване на договора	Окончателно приключване на работата по договора	<ul style="list-style-type: none"> • Телефон, електронна поща, работна среща; • Софтуер: MS Outlook; MS Word, MS Excel, MS Project, MS Visio, MS Power Point

4.2.6. Гарантиране и повишаване на качеството чрез предлаганите методи за изпълнение в предлаганата организация и методология за изпълнение на договора

Качеството на изпълнение на всяка дейност или процес е съвкупност от характеристики, които се прилагат с оглед постигане на заложените или по-високи цели. В настоящата обществена поръчка гарантирането на качеството на изпълнение се постига чрез прилагане на утвърдени добри практики и стандарти за изпълнение на договора, а повишаването на качеството се постига чрез адаптиране и комплексно комбиниране на доказани технологии и методологии за изпълнение на специфичните задачи в рамките на договора:



Фигура 2 Гарантиране и повишаване качеството чрез предлаганите методи за изпълнение

Основа за постигане на високо качество е прилагане на утвърдена методологична рамка за изпълнение на договора – PMBOK, RUP. PMBOK (Project Management Body of Knowledge – Система от знания за управление на проекти) за първи път е публикуван през 1996 г. от Института за управление на проекти (PMI – Project Management Institute). В течение на времето и прилагането му в практиката PMBOK претърпява шест изменения и се очаква във втората половина на 2017 г. да излезе седмото. Всички изменения в PMBOK показват практическата приложимост и доказано въздействие на системата от знания за постигане на успешен край и качество на проектите, изпълнявани по този стандарт. С прилагането му в настоящата обществена поръчка „Смарт Системс 2010“ ЕООД до голяма степен гарантира качеството на изпълнение на дейностите в обхвата на договора.

RUP (Rational Unified Process) е утвърдена итеративна рамка за създаване на софтуер. За първи път рамката е дефинирана като Rational Objectory Process (ROP) през 1996 г. и аналогично на PMBOK претърпява промени и усъвършенстване. RUP изпълнява следните стратегически насоки:

- Адаптиран процес за разработка;
- Инструменти, които автоматизират прилагането на процеса;
- Услуги, които ускоряват разбирането за процеси и инструменти.

Шестте най-добри практики в разработката на софтуер са дефинирани в RUP и включват:

- Итеративен процес на разработка;
- Управление на изискванията;

- Архитектура, базирана на компоненти;
- Визуално представяне на софтуера;
- Постоянна проверка на качеството;
- Управление на промените.

Прилагането на тези добри практики гарантират управление и проследяване на процеса по разработка на системата, което спомага за повишава качеството на крайния продукт.

Симбиозата между утвърдени стандарти, добри практики и процеси, изградени на базата на натрупан опит и постигнати успехи, са гаранция за взети мерки за гарантиране и повишаване на качеството на изпълнение.

В допълнение към описаните методологии и начин на прилагане, се наслагват методите за изпълнение, които включват техники и инструменти за прилагането им. Техниките и инструментите са практическите средства, посредством които се извършва работата. Аналогично на предложените методологии, „Смарт Системс 2010“ ЕООД предлага световно утвърдени техники (анализ, разработване, съгласуване, проследяване и контрол, докладване и т.н.) и инструменти (MS Office, MS Visio, т.н.), за да гарантира качествено изпълнение и постигане на максимални резултати.

Както се вижда от фигура „Гарантиране и повишаване качеството чрез предлаганите методи за изпълнение“, наслагването на методологична рамка, адаптиран начин на прилагане спрямо спецификата на проекта и дефиниране на конкретни методи за изпълнение повишават качеството и гаранцията за постигане на успешни резултати в края на проекта.

4.2.7. Обосновка на предлаганата методика за изпълнение и предлаганата организация за изпълнение към реализирането на резултатите от обществената поръчка

Описаните в настоящия документ методи за изпълнение на предлаганата организация и методология за изпълнение на договора са подбрани съобразно спецификата на обществената поръчка и стремежът за постигане на качество на изпълнението. Ползите и предимствата на предлаганата методология са детайлно изложени в т. 4.2.2. от настоящия документ. Предлаганата методология за управление на проекта РМВОК е сред най-прилаганите и доказани в практиката системи от знания за управление на проекти, прилагана в голяма част от проектите с подобен мащаб. РМВОК е напълно относима към предмета на обществената поръчка с дефинираните в нея основни процеси за управление на един проект:

- Процеси за инициране на проекта;
- Процеси за планиране;
- Процеси за изпълнение;
- Процеси за наблюдение и контрол;
- Процеси за приключване.

Предложената методологична рамка за изпълнение на договора, респективно прилаганите в нея методи за изпълнение, са относими към спецификата и изискванията на обществената поръчка и очакваните резултати от изпълнението, като за целта се прилага адаптиране на методологичната рамка към спецификата на изискванията на Възложителя.

Всяка методологична рамка се нуждае от техники, средства и инструменти (методи за изпълнение), за да може да бъде приложена и изпълнена на практика. Независимо колко е добра една идеология, е необходим най-малко молив, за да бъде описана. С оглед на това считаме, че описаните в настоящото предложение методи за изпълнение в предлаганата организация и методология за изпълнение на договора, са необходими и относими за качествено и в срок изпълнение на договора. Предложените методи за изпълнение са базирани на световно признати софтуерни продукти и средства за създаване на документи, продукти, анализиране и обобщаване на информация и изработване на отчетните резултати по изпълнение на договора.

4.3. Организация на екипа за изпълнение

За изпълнението на настоящата поръчка Сمارт Системс т разполага и ще осигури необходимите експерти, в съответствие с Техническите изисквания посочени от Възложителя. Експертите имат квалификация и правоспособност да изпълняват качествено, в срок и по обем всички задължения на Смарт Системс описани в Техническата спецификация на поръчката през целия срок на договора.

Минималният експертен състав за изпълнение на поръчката включва следните лица, които отговарят на посочените изисквания за професионална компетентност:

	Експерт (трите имена и длъжност в екипа)	Образование (степен, специалност, № на диплома, учебно заведение)	опит (предмет на дейността/услугата, период, длъжност, основни функции)
1.	Марин Викторов Кошутов – Ръководител на проекта	Бакалавър Компютърни системи и технологии Нов български университет, София Диплома № 212499, Серия НБУ – 2010, с регистрационен № 016002, от 2010 г., издадена от Нов български университет, София	Участие в разработка на внедрени проекти в подобна тематична област - разработване, внедряване и поддръжка на информационни системи и/или регистри и/или приложен софтуер. Успешно изпълнени проекти: Срок на проекта: 05.2014 – 12.2014 Проект „Проектиране, дизайн и програмиране на Интернет портал.“ Заемана позиция: Ръководител проект Отговорности и задължения: Изпълнява дейностите по управление на проекта, Изпълнител: Смарт системс 2010 ЕООД Възложител: Прокуратура „ИНФО КЛАС“ ООД Срок на проекта: 09.2014 – 09.2017

			<p>Проект: „Разработка и внедряване на уеб-базирана система за генериране на справки и отчети към хранилище за данни.“</p> <p>Заемана позиция: Ръководител проект Отговорности и задължения: Изпълнява дейностите по управление на проекта, Изпълнител: „Легит“ ЕООД Възложител: „Джойн Консултинг“ АД</p> <p>Срок на проекта: 04.2015 –09.2016</p> <p>Проект: Разработване, внедряване и гаранционна поддръжка на информационна система за управление на информация, свързана с хода на делата на „Водоснабдяване и канализация“ ЕАД.</p> <p>Заемана позиция: Ръководител проект Отговорности и задължения: Изпълнява дейностите по управление на проекта, Изпълнител: „ЛегИТ“ ЕООД Възложител: Водоснабдяване и канализация ЕАД</p> <p>Срок на проекта: 10.2017 – 10.2018</p> <p>Проект „Редизайн на УИС-2 в ПРБ с цел преминаване към използване на електронни документи в рамките на ПРБ и между ПРБ и др. държавни институции. Предоставяне на електронни услуги и информация за граждани.” Обособена позиция №1 „Въвеждане на електронен документооборот в Прокуратурата на РБ.“</p> <p>Заемана позиция: Ръководител проект Отговорности и задължения: Изпълнява дейностите по управление на проекта, Изпълнител: Гравис ЕООД Възложител: Прокуратура на Република България</p> <p>Срок на проекта: 05.2012 – 11.2012</p> <p>Проект „Разширяване на функционалните възможности на Автоматизираната система за управление на дела, внедрена във Върховния касационен съд, както и на модула за публикуване на информация от делата в официалния сайт на съда, с цел реализация на модул за разширено</p>
--	--	--	---

			<p>търсене на съдебни актове по материя към уебсайта на ВКС.“</p> <p>Заемана позиция: Ръководител проект</p> <p>Отговорности и задължения: Изпълнява дейностите по управление на проекта,</p> <p>Изпълнител: Гравис ЕООД</p> <p>Възложител: Върховен касационен съд</p>
2	Владимир Радославов Маринов – Системен архитект	<p>Бакалавър</p> <p>Телекомуникации</p> <p>Комуникационна техника и технологии</p> <p>Технически университет София</p> <p>Проф. квалификация: инженер по комуникации</p> <p>Диплома Серия ТУ-СФ - 2006 №000919, издадена от Технически университет София</p>	<p>Участие като системен архитект в реализацията на завършени проекти, свързани с разработка, внедряване и поддръжка на информационни технологии и/или информационни системи и/или информационни услуги.</p> <p>Успешно изпълнени проекти:</p> <p>Срок на проекта: 09.2014 – 09.2017</p> <p>Проект: „Разработка и внедряване на уеб-базирана система за генериране на справки и отчети към хранилище за данни.“</p> <p>Заемана позиция: Системен архитект</p> <p>Отговорности и задължения: Участва в дейностите по създаване на архитектури на информационната система по утвърден модел.</p> <p>Изпълнител: „Легит“ ЕООД</p> <p>Възложител: „Джойн Консултинг“ АД</p>
3	Тодор Велев Велев - Бизнес анализатор	<p>Магистър</p> <p>„Системи и Управление“</p> <p>Технически Университет – София</p> <p>Диплома №001372 , Серия А-95 ТУ-Сф.</p>	<p>Участие като бизнес анализатор в реализацията на завършени проекти, свързани с разработка, внедряване и поддръжка на информационни технологии и/или информационните системи и/или информационните услуги.</p> <p>Успешно изпълнени проекти:</p> <p>Срок на проекта: 2016 - 2017</p> <p>Проект: „Проектиране, разработка и внедряване в експлоатация на Софийски общински туристически регистър (с публична УЕБ базирана част) и УЕБ базирана система „Административен портал“ (www.info-sofia.bg) за нуждите на общинско предприятие „Туризм“</p> <p>Заемана позиция: Бизнес анализатор</p> <p>Отговорности и задължения: Участва в дейностите по описание на бизнес процесите, бизнес анализ, бизнес решения</p>

			<p>и моделиране на процеси в областта на информационни системи и технологии. Изпълнител: Перфект Плюс ЕООД Възложител: ОП Туризм</p> <p>Срок на проекта: 09.2014 – 09.2017 Проект: „Разработка и внедряване на уеб-базирана система за генериране на справки и отчети към хранилище за данни.“ Заемана позиция: Бизнес анализатор Отговорности и задължения: Участва в дейностите по описание на бизнес процесите, бизнес анализ, бизнес решения и моделиране на процеси в областта на информационни системи и технологии. Изпълнител: „Легит“ ЕООД Възложител: „Джойн Консултинг“ АД</p>
4	<p>Стоянка Атанасова Мишева Старши програмист</p> <p>–</p>	<p>Магистър „Компютърни системи“, Диплома No002134 от 2002 г. Технически университет –София.</p>	<p>Участие в проектиране, разработка и внедряване на проекти в областта на информационните технологии и/или информационните системи и/или информационните услуги.</p> <p>Успешно изпълнени проекти: Срок на проекта: 01.2016 – 01.2017 Проект: „Проектиране, разработка и внедряване в експлоатация на Софийски общински туристически регистър (с публична УЕБ базирана част) и УЕБ базирана система „Административен портал“ (www.info-sofia.bg) за нуждите на общинско предприятие „Туризм“ Заемана позиция: Програмист Отговорности и задължения: Участва в дейностите по проектиране, разработка, внедряване и поддръжка на информационната система. Изпълнител: Перфект Плюс Възложител: ОП „Туризм“</p> <p>Срок на проекта: 02.2015 – 02.2018 Проект: Проектиране, разработка, внедряване и поддръжане на „Национален туристически регистър“ Заемана позиция: Програмист Отговорности и задължения: Участва в дейностите по проектиране, разработка,</p>

			внедряване и поддръжка на информационната система. Изпълнител: Перфект Плюс Възложител: Министерство на туризма;
5	Искрен Илиев Иванов – Старши програмист	Бакалавър Компютърни системи и технологии Технически университет Варна Диплома № 040495, Серия ТУВн – 2009, издадена на 16.04.2009 г, издадена от Технически университет Варна	<p>Участие в проектиране, разработка и внедряване на проекти в областта на информационните технологии и/или информационните системи и/или информационните услуги.</p> <p>Успешно изпълнени проекти: Срок на проекта: 05.2016 – 04.2017 Проект „Разработване на допълнителни функционалности и поддръжка на съществуващи функционалности в уеб-базираната Унифицирана информационна система 2 (УИС2) “. Заемана позиция: Програмист Отговорности и задължения: Участва в дейностите по проектиране, разработка, внедряване и поддръжка на информационната система. Изпълнител: Гравис ЕООД Възложител: Прокуратура на Република България</p> <p>Срок на проекта: 04.2015 – 03.2016 Проект „Разработване на функционалности на уеб-базираната Унифицираната информационна система на Прокуратурата на Република България”. Заемана позиция: Програмист Отговорности и задължения: Участва в дейностите по проектиране, разработка, внедряване и поддръжка на информационната система. Изпълнител: Гравис ЕООД Възложител: Прокуратура на Република България</p> <p>Срок на проекта: 10.2017 – 10.2018 Проект „Редизайн на УИС-2 в ПРБ с цел преминаване към използване на електронни документи в рамките на ПРБ и между ПРБ и др. държавни институции. Предоставяне на електронни услуги и информация за граждани.” Обособена</p>

			<p>позиция №1 „Въвеждане на електронен документооборот в Прокуратурата на РБ.“</p> <p>Заемана позиция: Програмист</p> <p>Отговорности и задължения: Участва в дейностите по проектиране, разработка, внедряване и поддръжка на информационната система.</p> <p>Изпълнител: Гравис ЕООД</p> <p>Възложител: Прокуратура на Република България</p>
6	<p>Антонио Бисеров – Петров Програмист</p>	<p>Бакалавър Електроника Технически университет, Варна диплома № 275432, Серия ТУВн – 2015 г., издадена от Технически университет, Варна</p>	<p>Участие в проектиране, разработка и внедряване на проекти в областта на информационните технологии и/или информационните системи и/или информационните услуги.</p> <p>Успешно изпълнени проекти: Срок на проекта: 10.2017- 10.2018 Проект: Редизайн на УИС-2 в ПРБ с цел преминаване към използване на електронни документи в рамките на ПРБ и между ПРБ и др. държавни институции.</p> <p>Предоставяне на електронни услуги и информация за граждани.“ по Обособена позиция № 2: „Предоставяне на електронни услуги за граждани, фирми, държавни институции и Единната информационна система за противодействие на престъпността“</p> <p>Заемана позиция: Програмист</p> <p>Отговорности и задължения: Участва в дейностите по проектиране, разработка, внедряване и поддръжка на информационната система.</p> <p>Изпълнител: „Легит“ ЕООД</p> <p>Възложител: „Сирма Солюшънс“ АД</p> <p>Срок на проекта: 08.2017- 06.2018 Проект: „Разработване и внедряване на специализиран софтуер за мониторинг, оценка, прогнозиране на метеорологични условия във ветро парковете и автоматизиране на процеси в управлението на съществуващите системи“</p> <p>Заемана позиция: Програмист</p> <p>Отговорности и задължения: Участва в дейностите по проектиране, разработка,</p>

			<p>внедряване и поддръжка на информационната система. Изпълнител: „Легит“ ЕООД Възложител: „ВИП КАПИТАЛ“ ООД</p> <p>Срок на проекта: 09.2014 – 09.2017 Проект: „Разработка и внедряване на уеб-базирана система за генериране на справки и отчети към хранилище за данни.“ Заемана позиция: Програмист Отговорности и задължения: Участва в дейностите по проектиране, разработка, внедряване и поддръжка на информационната система. Изпълнител: „Легит“ ЕООД Възложител: „Джойн Консултинг“ АД</p>
8	Дарин Емилов - Железов Програмист	Магистър „Информационни технологии“ Диплома №236915 от 2015, УниБИТ – София.	<p>Участие в проектиране, разработка и внедряване на проекти в областта на информационните технологии и/или информационните системи и/или информационните услуги.</p> <p>Успешно изпълнени проекти: Срок на проекта: 07.2011 г. - 12.2013 г. Проект: „Обновяване на информационната система за производството по несъстоятелност, интеграция с други системи, организация на дейността по въвеждане на данни и съпътстващо обучение” Заемана позиция: Програмист Отговорности и задължения: Участва в дейностите по проектиране, разработка и внедряване на информационната система. Изпълнител: Перфект Плюс Възложител: Министерство на туризма;</p>
9	Димитър Филипов Миладинов – Програмист интеграции	Магистър, Технология на материалите и материалознание, Диплома с рег. № 021252/1998 г, Химико технологичен и металургичен университет – София	<p>Участие в интеграционни проекти в областта на информационните технологии и/или информационните системи и/или информационните услуги.</p> <p>Успешно изпълнени проекти Срок на проекта: 07.2012 – 05.2014 Интеграционен проект: Доизграждане и усъвършенстване на ЕИСПП и</p>

			<p>интегриране на ведомствените информационни системи с ядрото на ЕИСПП Обособена позиция № 7:</p> <p>„Надграждане и свързване на съществуващата деловодна информационна система от типа ДИС (Деловодна информационна система) на Военно-апелативен съд София и военните съдилища в гр. София и Плевен, включително разработване и изграждане на конвертор и филтри към нея, като елемент на ведомствени комуникационни компоненти, съобразно чл. 379, ал. 2 от ЗСВ при регистриране на събития и обекти по наказателния процес към ядрото на ЕИСПП</p> <p>Роля в проекта: Програмист интеграции.</p> <p>Отговорности и задължения: Извършване на интеграционни услуги, надграждане и свързване на съществуващата деловодна информационна система от типа ДИС (Деловодна информационна система) на Военно-апелативен съд София и военните съдилища в гр. София и Плевен, включително разработване и изграждане на конвертор и филтри към нея, при регистриране на събития и обекти по наказателния процес към ядрото на ЕИСПП</p> <p>Срок на проекта: 07.2012 – 05.2014</p> <p>Интеграционен проект: Доизграждане и усъвършенстване на ЕИСПП и интегриране на ведомствените информационни системи с ядрото на ЕИСПП”, Обособена позиция № 6:</p> <p>„Надграждане и свързване на съществуващата деловодна информационна система от типа АСУД (Автоматизирана информационна система за управление на делата), включително разработване и изграждане на конвертор и филтри към нея, като елемент на филтри към нея, като елемент на ведомствени комуникационни компоненти, съобразно чл. 379, ал. 2 от ЗСВ при регистриране на събития и обекти по наказателния процес към ядрото на ЕИСПП</p>
--	--	--	---

			<p>Роля в проекта: Програмист интеграции.</p> <p>Отговорности и задължения: Извършване на интеграционни услуги, надграждане и свързване на съществуващата деловодна информационна система от типа АСУД (Автоматизирана информационна система за управление на делата), включително разработване и изграждане на конвертор и филтри към нея, като елемент на филтри към нея, като елемент на ведомствени комуникационни компоненти, съобразно чл. 379, ал. 2 от ЗСВ при регистриране на събития и обекти по наказателния процес към ядрото на ЕИСПП</p> <p>Срок на проекта: 04.2015 –09.2016</p> <p>Проект: Разработване, внедряване и гаранционна поддръжка на информационна система за управление на информация, свързана с хода на делата на „Водоснабдяване и канализация“ ЕАД.</p> <p>Заемана позиция: Програмист интеграции.</p> <p>Отговорности и задължения: Участва в дейностите по проектиране, разработка и внедряване на информационната система, осъществяване на интеграция между информационната система и съществуващата система за управление на плащанията.</p> <p>Изпълнител: „ЛегИТ“ ЕООД</p> <p>Възложител: Водоснабдяване и канализация ЕАД</p>
10	Кирил Руменов Илиев - Контрол на качеството	<p>Бакалавър</p> <p>Информационни технологии</p> <p>Университет по библиотекознание и информационни технологии София</p> <p>Издадена диплома с номер № 293587, Серия УниБИТ-2017, издадена на 09.10.2017 г. от Университет по библиотекознание и</p>	<p>Участие като специалист по качество в реализацията на завършени проекти, свързани с проектиране, разработване, внедряване и поддръжка на информационните технологии и/или информационните системи и/или информационните услуги.</p> <p>Успешно изпълнени проекти:</p> <p>Срок на проекта: 04.2015 –09.2016</p> <p>Проект: Разработка на уеб-базирана приложна информационна система за автоматизация на бизнес процеси в „Сервие Медикал” ЕООД</p>

		<p>информационни технологии София</p>	<p>Заемана позиция: Специалист осигуряване на качеството</p> <p>Отговорности и задължения: Участва в следните дейности по осигуряване на качеството при проектиране, разработване, внедряване и поддръжка на уеб-базираната приложна софтуерна система. Извършвайки следните дейности:</p> <ul style="list-style-type: none"> • Изпитване на софтуера • Създаване на потребителски случаи • Автоматично тестване <p>Изпълнител: „ЛегИТ“ ЕООД Възложител: Сервие Медикал</p> <p>Срок на проекта: 10.2017- 10.2018 Проект: Редизайн на УИС-2 в ПРБ с цел преминаване към използване на електронни документи в рамките на ПРБ и между ПРБ и др. държавни институции. Предоставяне на електронни услуги и информация за граждани.“ по Обособена позиция № 2: „Предоставяне на електронни услуги за граждани, фирми, държавни институции и Единната информационна система за противодействие на престъпността“</p> <p>Заемана позиция: Специалист осигуряване на качеството</p> <p>Отговорности и задължения: Участва в следните дейности по осигуряване на качеството при проектиране, разработване, внедряване и поддръжка на уеб-базираната приложна софтуерна система. Извършвайки следните дейности:</p> <ul style="list-style-type: none"> • Изпитване на софтуера • Създаване на потребителски случаи • Автоматично тестване <p>Изпълнител: „Легит“ ЕООД Възложител: „Сирма Солюшънс“ АД</p>
--	--	---------------------------------------	--

4.3.1.1. Роли и отговорности на членовете на екипа на Сمارт Системс

За успешната реализация на настоящата обществена поръчка от съществено значение е своевременното осигуряване на необходимата информация в правилния формат, в точното време и с точното въздействие на участниците по проекта, на които е необходима.

Таблица 4 - Роли и отговорности на членовете на екипа на Сمارт Системс

Ключов експерт	Роля и отговорности за изпълнение на задачите от проектната документация	Изпълнител на ролята
Ключов експерт №1: „Ръководител на проекта“	<p>Методологията RUP дефинира обхвата от отговорностите на Ръководителя за изпълнение на проекта в следния ред – хора, продукт, процес, проект. Събирателният артефакт/документ за тези четири елемента е Планът на проекта, затова и той ще бъде настолната книга или Библията за Ръководителя на проекта.</p> <p>Основни задължения:</p> <ul style="list-style-type: none"> • отговаря за ефективното и качествено управление и изпълнение на обществената поръчка като ръководи изпълнението на дейностите; • организира и координира цялостната дейност на екипа за изпълнение на поръчката; • осъществява контакт с органи и институции, имащи отношение към изпълнението на настоящата поръчка и проекта като цяло по отношение на изготвяне на информационната система; • организира и координира изпълнението на предписанията на Възложителя; • изготвя докладите предмет на договора за обществена поръчка; • при откриване/докладване на нередност, или подозрение за нередност или измама на 	<p>За Ръководител на проекта предлагаме:</p> <p>Марин Викторов Кошутов</p>

	който и да било етап от изпълнението на настоящата поръчка и проекта, докладва на Възложителя - НСИ.	
Ключов експерт № 2: „Системен архитект“	<p>проследява реализацията на изготвените спецификации на системни изисквания;</p> <ul style="list-style-type: none"> • Следи за прилагане на избраните стандарти и методи за разработка, в зависимост от характеристиките на системата и останалите приложими стандарти и изисквания на Възложителя; • Следи за качеството на софтуера; • Следи за правилното имплементиране на взаимовръзките и зависимостите между компонентите; • Анализира изискванията по отношение на дизайна на базата данни и имплементира промени в базата при необходимост; • Проектира, разработва и имплементира структури на бази данни; • Изготвя документация, свързана с дизайна и описанието на базата данни; • Разработва методологии, справки, заявки и репликация на таблици, като следи всички данни да са в правилния формат; • Участва в подготовката на тестова среда и провеждане на системни тестове; • Проследява изискванията към реализацията на системната архитектура; • Участва при въвеждане на промените на софтуера в тестова и/или продуктивна среда и при провеждане на тестовите изпитания; • Разработва техническа документация по проекта. 	<p>За Системен архитект предлагаме:</p> <p>Владимир Радославов Маринов</p>
Ключов експерт № 3: „Бизнес анализатор“	<ul style="list-style-type: none"> • Анализират процедурите и работните процеси. 	За Бизнес анализатор предлагаме:

	<ul style="list-style-type: none"> • Анализират необходимостта и възможните връзки за интеграция. • Оптимизират и моделират бизнес процеси, чрез софтуер за моделиране на процеси; • Определят и проследяват съответствията с бизнес целите, предмета и дефинираните изискванията; • Участват в изграждането на цялостната концепция за архитектурата на системата; • Изготвят детайлна техническа спецификация. • Създават за всеки процес графично и текстово описание. • Участват в подготовката и провеждането на различни форми за обучение на потребителите; • Участват в процеса по отстраняване на откритите дефекти по време на тестването, експлоатацията на системите и гаранционната поддръжка на системите; • Изготвят документация по извършените дейности: спецификации, инструкции, ръководства за потребителите, указания, помощни материали и друга техническа документация, съгласно изискванията на Възложителя. 	Тодор Велев Велев
Ключов експерт №3: „Старши Програмист“	Взаимодейства с всички специалисти по изпълнение на обществената поръчка за постигане на общите цели при реализацията на проекта;	За Старши Програмист предлагаме: Стоянка Атанасова Мишева

	<p><u>Основни задължения:</u></p> <ul style="list-style-type: none"> • Участват в подготвянето на план за разработване на софтуера; • Разработват прототипи за потвърждаване на изискванията; • Разработват моделите на базите данни; • Разработват компоненти; • Контролират качеството на изходния код (code review); • Документират разработения софтуер; • Участват в разработването на тестови сценарии; • Участват в подготовката на тестова среда и провеждане на вътрешни системни тестове. 	
<p>Ключов експерт №4: „Старши Програмист“</p>	<p>Взаимодейства с всички специалисти по изпълнение на обществената поръчка за постигане на общите цели при реализацията на проекта;</p> <p><u>Основни задължения:</u></p> <ul style="list-style-type: none"> • Участват в подготвянето на план за разработване на софтуера; • Разработват прототипи за потвърждаване на изискванията; • Разработват моделите на базите данни; • Разработват компоненти; • Контролират качеството на изходния код (code review); • Документират разработения софтуер; 	<p>За Старши Програмист предлагаме:</p> <p>Искрен Илиев Иванов</p>

	<ul style="list-style-type: none"> • Участват в разработването на тестови сценарии; • Участват в подготовката на тестова среда и провеждане на вътрешни системни тестове. 	
Ключов експерт №6: „Програмист“	<p>Взаимодейства с всички специалисти по изпълнение на обществената поръчка за постигане на общите цели при реализацията на проекта;</p> <p><u>Основни задължения:</u></p> <ul style="list-style-type: none"> • Участват в подготвянето на план за разработване на софтуера; • Разработват прототипи за потвърждаване на изискванията; • Разработват моделите на базите данни; • Разработват компоненти; • Контролират качеството на изходния код (code review); • Документират разработения софтуер; • Участват в разработването на тестови сценарии; • Участват в подготовката на тестова среда и провеждане на вътрешни системни тестове. • 	<p>За Програмист предлагаме:</p> <p>Антонио Бисеров Петров</p>
Ключов експерт №7: „Програмист“	<p>Взаимодейства с всички специалисти по изпълнение на обществената поръчка за постигане на общите цели при реализацията на проекта;</p> <p><u>Основни задължения:</u></p>	<p>За Програмист предлагаме:</p> <p>Дарин Емилов Железов</p>

	<ul style="list-style-type: none"> • Участват в подготвянето на план за разработване на софтуера; • Разработват прототипи за потвърждаване на изискванията; • Разработват моделите на базите данни; • Разработват компоненти; • Контролират качеството на изходния код (code review); • Документират разработения софтуер; • Участват в разработването на тестови сценарии; • Участват в подготовката на тестова среда и провеждане на вътрешни системни тестове. • Отстранява грешки открити при провеждане на тестове на системата 	
Ключов експерт №8: „Програмист интеграция“	<p>Взаимодейства с всички специалисти по изпълнение на обществената поръчка, както и с други служители на Сمارт Системс за постигане на общите цели при реализацията на проекта;</p> <p><u>Основни задължения:</u></p> <ul style="list-style-type: none"> • Участват в подготвянето на план за разработване на софтуера; • Разработват прототипи за потвърждаване на изискванията; • Разработват моделите на базите данни; • Разработват компоненти; • Контролират качеството на изходния код (code review); 	<p>За Програмист интеграция предлагаме:</p> <p>Димитър Филипов Миладинов</p>

	<ul style="list-style-type: none"> • Документират разработения софтуер; • Участват в разработването на тестови сценарии; • Участват в подготовката на тестова среда и провеждане на вътрешни системни тестове. 	
Ключов експерт №9: „Контрол на качеството“	<p>Има предимно проактивни, организаторски, превантивни и контролни функции по отношение на цялостното прилагане на методиката по управление на качеството, описана в „План за управление на качеството“;</p> <p>Взаимодейства с всички специалисти по изпълнение на обществената поръчка, както и с други служители на Сمارт Системс за постигане на общите цели при реализацията на проекта;</p> <p>Следи прилагането на стандартите за качество и спазване на изискванията на обществената поръчка;</p> <p>Следи за изпълнението на документа „План за управление на качеството“;</p> <p>Следи за правилното прилагане на изискванията</p> <p>Интегрираната система за управление на качеството, информационната сигурност и услугите, внедрена в съответствие с ISO 9001:2015 и ISO/IEC 27001:2013;</p> <p><u>Основни задължения:</u></p> <ul style="list-style-type: none"> • Разработват спецификации за тестване; • Участват в създаването на симулации и прототипи за потвърждаване на изискванията; • Проследяват тестовите приложения в съответствие с функционалните изисквания; 	<p>За Контрол на качеството предлагаме:</p> <p>Кирил Руменов Илиев</p>

4.3.1.2. Структура на екипа на Сمارт Системс

Разработката на софтуер е била и ще бъде силно зависима от субективния фактор – опита и знанията на експертите, участващи в проекта. И естествено, причината за това е, че разработката на софтуер е високо интелектуална дейност.

Ето защо, създаването на правилна вътрешна организация за изпълнение на проекта и точното дефиниране и разпределение на ролите и отговорностите на ключовите експерти от екипа за изпълнение при Сمارт Системс са най - важните предпоставки за успешното и качествено реализиране на всеки ИТ проект.

Във връзка с горното и отчитайки опита на „Смарт Системс 2010“ ЕООД предлагаме следната организация за изпълнение на дейностите по проекта по роли и отговорности за изпълнение на задачите от Техническата спецификация, базирана на принципите и най-добрите практики на RUP за прозрачност, приемственост и възможност за обмен на ноу-хау между експертите, участващи в изпълнението на проекта.



Фигура 3 Организационната структура (органиграма) на Екипа за изпълнение на проекта при Смарт Системс

4.3.1.3. Начин на взаимодействие между членовете на екипа на Смарт Системс ;

Начините на тази комуникация ще бъдат:

- провеждане на регулярни срещи на екипа - по време на регулярните срещи ще се разпределят съответните отговорности, задължения за всеки член на екипа, а от друга страна, ще се търси обратна връзка за постигнатите резултати. По време на провежданите срещи ще се осъществява мониторинг на изпълнението от страна на всеки участник в екипа по проекта, отчитане на напредъка и статуса на изпълнение на задачите;
- срещи по конкретни теми - ще бъдат организирани оперативни срещи по конкретни теми, при възникване на необходимост от изясняване на конкретна тема или при възникнал проблем, изискващ допълнително обсъждане; упълномощеното по договора лице от страна на „Смарт Системс 2010“ ЕООД ще отговаря достигналата до него информация по договора да бъде своевременно предоставена (устно, по телефон, по електронна поща, на провежданите регулярни срещи) на участниците в екипа, на които е необходима.

4.3.1.4. Връзки за взаимодействие с екипа на Възложителя

Външна комуникация между упълномощените за това лица, представители на „Смарт Системс 2010“ ЕООД и Възложителя:

Предвид характера и мащаба на предвидения за изпълнение договор по настоящата обществена поръчка и отчитайки изискванията на проекто-договора, който е част от документацията по поръчката, предлагаме всички съобщения и уведомления между страните по договора да се отправят в писмена форма от упълномощените за това лица на страните по него. При стартиране на дейностите по договора Възложителят и „Смарт Системс 2010“ ЕООД определят упълномощените за това лица, през които преминава цялата комуникация по договора, а те от своя страна имат ангажимента да предоставят необходимата информация на заинтересованите страни вътре в организацията си. При провеждането на въстъпителната среща по договора ще бъдат уточнени упълномощените лица на страните, адресите за кореспонденция и други детайли като телефонни номера, e-mail.

Писмената форма ще се счита за спазена и когато съобщенията и уведомленията са отправени по факс, електронна поща или чрез лицензирани/регистрирани пощенски оператори.

Правилата за провеждане на срещи и за използване на другите комуникационни канали, които ще бъдат съгласувани заедно с представители на Възложителя. Срещите могат да се инициират от Смарт Системс или от Възложителя.

Планиране и подготовка

- Всяка среща трябва да има предварително ясна за всички участници цел;
- Подборът и броя на участниците трябва да съобразен с темите на срещата;

- Мястото, началният час и планираната продължителност трябва да са съобщени възможно най-рано;
- До всички участници трябва да се изпратят материалите, с които трябва да се запознаят преди срещата, като колкото по-голям е обемът им, толкова повече време преди срещата трябва да са разпратени;
- Ако са необходими проектор, флипчарт и друго оборудване, тази информация трябва да се съдържа в съобщението за срещата;
- Подготовка на презентации (ако такива се предвиждат);
- Действия непосредствено преди срещата (предишния ден).

Протоколиране и регистриране

Всички срещи трябва да бъдат протоколирани във формат на възприет шаблон и като минимум ще се отразят:

- Място на провеждане;
- Продължителност;
- Основна тема;
- Инициали на участниците;
- Взети решения;
- Планирани действия с отговорници и срокове;
- Съпътстващи документи под формата на приложения в случай на необходимост.

Работните срещи се протоколират от Сمارт Системс и протоколите се изпращат на Възложителя за сведение и съгласуване, а само регистрирани, тогава, когато информацията от тези срещи ще бъде документирана в някой от продуктите (резултатите) на проекта и участниците в срещата ще участват във валидирането му.

Срещите в настоящия проект могат да се класифицират на:

- Контролни;
- Работни;
- Срещи за представяне на резултати.

За разпространението на информацията между участниците в проекта и другите заинтересовани лица ще бъдат използвани следните комуникационни средства:

- Електронна поща – e-mail кореспонденцията ще бъде използвана за ежедневна комуникация и разпространение на информация. Това средство за комуникация ще бъде използвано за разпространение на оперативни документи между членовете на екипите, ръководителите на проекта и другите заинтересовани лица. Документи, изискващи одобрение, се изпращат първо по електронна поща за съгласуване с другата страна, след което се разпечатват на хартия и се подписват. Чрез електронна поща се насрочват и срещите по проекта, независимо от техния характер;

- Хартиен носител – на хартиен носител ще бъдат разпечатвани и разпространявани документите, които са резултати от изпълнението на проекта и такива, които изискват одобрение: доклади, планове, протоколи, спецификации и др.;

- Работният език за всички документи, които представляват официален продукт – предмет на приемане и предаване, е български. Допуска се използване на помощни документи на английски език (като например: технически описания на стандартни продукти);

- Телефон и Факс.

На тези срещи с Възложителя ще се разглеждат и разрешават всички въпроси, възникнали в процеса на работа. На тях ще се изслушва текущото състояние и напредъка от страна на „Смарт Системс 2010“ ЕООД, и ще се взимат решения от страна на Възложителя, съвместно с Смарт Системс по възникнали въпроси, документации или събития.

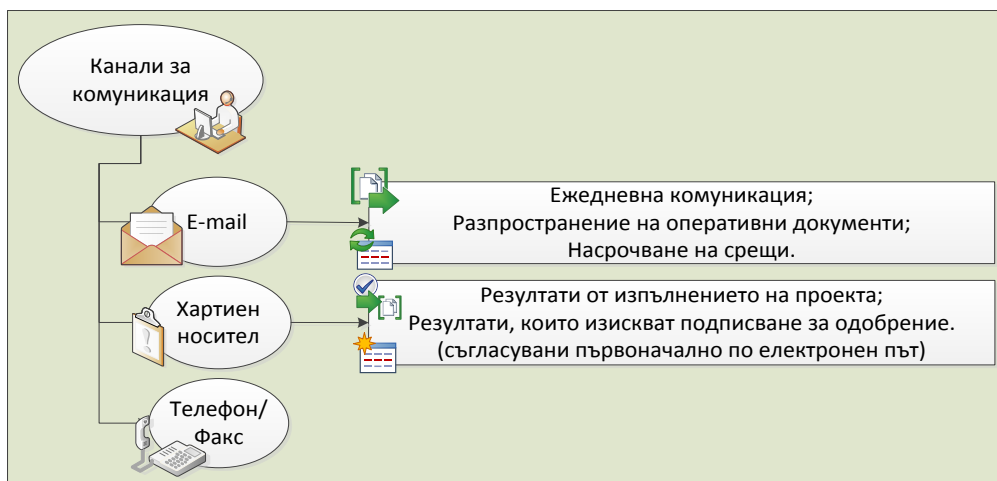
При необходимост ще бъдат и инициирани и срещи по конкретни теми, при възникване на необходимост от изясняване на конкретна тема или при възникнал проблем, изискващ допълнително обсъждане от двете страни.

В случай на необходимост в хода на изпълнение на проекта, ключовите експерти на Смарт Системс ще могат да осъществяват директна комуникация с ключови експерти на Възложителя. Този вид взаимодействие ще се извършва със съгласието на Ръководителя на проекта от страна на „Смарт Системс 2010“ ЕООД и Възложителя и по точно определени теми и/или въпроси, които изискват специализирани познания или професионални компетенции за извършване на конкретна дейност от проекта.

Всички отчетни документи и аналитични продукти в резултат на изпълнение на договора в редактируема електронна форма („.doc“ или „.docx“ файлов формат) ще подлежат на предварително съгласуване и одобрение по електронна поща от страна на Възложителя, преди официалното им изпращане на хартиен носител.

С цел осъществяване на добра координация между Смарт Системс и Възложителя, „Смарт Системс 2010“ ЕООД ще осигури екип за административно и експертно взаимодействие, както и нарочен технически сътрудник на проекта, който се грижи за цялостното документиране на преписките и документите в процеса на комуникация между страните.

Каналите на комуникация и вида на обменяната информация между страните при изпълнение на договора, схематично са визуализирани на следващата фигура.



Фигура 4 Канали за комуникация

4.4. Проектна документация

Подробно описание на проектната документация, видове доклади, техническа и експлоатационна документация, време на предаване, съдържание на документите, управление на версиите и др, може да бъде намерено в т. 9 „Документация“ от Техническото предложение.

4.5. Управление на качеството

4.5.1. Дефиниция и цел

Дефиниция

Съгласно ISO под качество се разбира степента на съответствие на крайните резултати от дейността на организациите с тези, определени като оптимални за задоволяването на необходимостта на потребителите, опазващи тяхното здраве и гарантиращи хармонията с околната среда.

В конкретния случай - разработване на информационна система, целта е резултата да съответства на предварително фиксирани изисквания (функционални и нефункционални), както и да удовлетворява идентифицираните потребителски нужди.

Качеството е най-важният аспект при проектирането и разработването на информационна система и представлява обобщаващ критерий за изпълнението на изискванията и целите на проекта, както и за ефективността на вложените в него средства.

Управлението на качеството представлява процес, включващ управленските функции планиране и взимане на решения, организиране, ръководене, координиране, мотивиране и контрол насочени към дейностите и ресурсите в организацията с цел постигане на пълно съответствие между крайните резултати от нейната дейност и идентифицираните изисквания за целта.

За постигане на очакваните крайни резултати от изпълнението на договорните отношения, освен контрол от страна на Възложителя, се извършва процес на вътрешно управление на качеството и организация на дейностите от страна на Сمارт Системс .

Управлението на качеството включва изискванията и процесите, които осигуряват успешното изпълнение на проекта. Всички дейности по управлението на проекта и крайният резултат се включват в мероприятията по осигуряване на качеството.



Фигура 5 Модел на СУК, основана на процеси

Цел

Целта на управлението на качеството е да се осигури краен продукт или услуга в рамките на очакванията на клиента, като начинът на изпълнение също може да бъде предварително определен и в последствие съблюдаван по време на изпълнението на проекта.

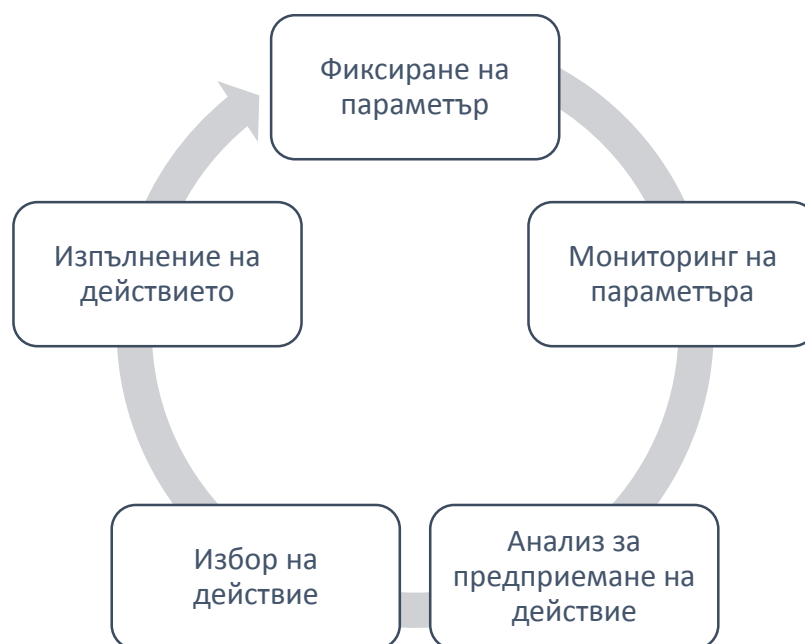
Независимо от използвания инструментариум, информацията, която се обобщава от мониторинга на качествените параметри, след съответния анализ, проследяване на тенденции и осмисляне в различен контекст може да доведе до:

- **Натрупване на допълнителна информация** с цел анализ и бъдещи действия при изпълнението на подобни проекти или дейности по проекти;
- **Коригиращи действия** - Това са целенасочени действия, които трябва да се предприемат, така че да се промени текущия процес по изпълнение на дейностите по проекта, така че изпълнението да се върне в рамките на предварително фиксираните параметри. Тези действия се предприемат, когато има промяна в очакваните (там където могат да се дефинират) ключови параметри свързани с качеството.
- **Превантивни мерки** - Изпълнението на целенасочените действията по превантивни мерки е подобно на коригиращите действия с тази разлика, че те ще се изпълняват преди да се установи „официално“ отклонение. Допълването на посочения по-горе пример е, че ако при изпълнението на дейностите се направи

предварително валидиране с Възложителя, тези отклонения ще се идентифицират в по-ранен момент и за бъдещи периоди ще може да се избегнат, като Сمارт Системс т извърши нужните промени предварително. Предприемането на такива мерки е сложен процес, който изисква наличието на сериозен опит или по-дейно участие на Възложителя и/или заинтересованите страни, което не винаги е осъществимо. Примери за случаи на предприети превантивни мерки са:

- Коригиране на работни процедури за конкретна дейност с цел да се предотврати появата на често случващ се проблем;
- Допълване на екипа по проекта.
- **Поправка на дефект** - Този подход се отнася за случаите при изработването на конкретен продукт или дизайн на често повтаряща се идентична услуга (най-често свързано с поддръжка на ИТ ресурс). Тогава се предприемат действията по корекция на изработеното, въз основа на резултата от тестването;
- **Обновяване (ъпдейт) на различни аспекти на проекта** - В този случай става дума за корекция на някои от планираните подходи за изпълнение на дейностите по проекта или други аспекти на проекта (примерно график), така че да се отразят модифицирани или ново-възникнали идеи, да се включат допълнителни параметри или да се модифицират съществуващи. На практика това може да се разгледа като частен случай на коригиращите действия или превантивните мерки, свързан с промяна в документ. Примери за такива обновявания в различна степен на значимост може да са обновявания в:
 - График за изпълнение на конкретна дейност;
 - Вътрешна (за „Смарт Системс 2010“ ЕООД) процедура за изпълнението на определена дейност по проекта;
 - Описание на критерии за приемането на дадена дейност.

Последователността от тези действия може да се изобрази схематично по следния начин:

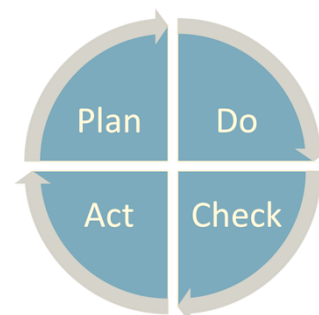


Фигура 6 Корекция/добавяне на параметър

4.5.2. Елементи на подхода за управление на качеството

Модерното управление на качеството е допълнение към управлението на проектите. И двете дисциплини подчертават важността на:

- **Удовлетворението на клиента** - Разбиране, оценяване, определяне и управление на очакванията, така че изискванията на клиента да бъдат изпълнени. Това налага едновременно придържане към изискванията (за да бъде сигурно, че проектът ще произведе това, за което е създаден) и пригодност за употреба (продуктът или услугата трябва да удовлетворяват реални потребности).
- **Превенцията вместо инспекцията** - Един от фундаменталните постулати на съвременното управление на качеството гласи, че качеството се планира, проектира, вгражда, но не и инспектира. Разходите за предотвратяване на грешките са в общия случай много по-ниски, отколкото разходите за коригиране на грешките, открити при инспекция.
- **Продължаващото усъвършенстване** - Цикълът „планирай-направи-провери-действай“ (Plan-Do-Check-Act - PDCA) е в основата на управлението на качеството, дефиниран от Шуърт и модифициран от Деминг. В допълнение инициативите за подобряване на качеството, предприемани от изпълняващата организация би следвало да подобрят качеството както на управлението на проекта, така и на качеството на продукта.



- **Отговорност на ръководството** - Постигането на успех изисква участието на всички членове на екипа по проекта, но отговорност единствено на ръководството е предоставянето на необходимите ресурси за този успех.

Управлението на качеството в един проект включва процесите и дейностите, които определят политиките по качеството, целите и отговорностите, така че проектът да удовлетвори нуждите, заради които е бил стартиран. Предлагаият от нас процес по управление на качеството при изпълнение на настоящата обществена поръчка е визуализиран на фигурата.



Фигура 7 Процес по управление на качеството

4.5.3. Измерване и осигуряване на качеството

Осигуряването на качеството е процесът на проверка на изискванията за качество. Дейностите по осигуряване на качеството често пъти се наблюдават от отдел по осигуряване на качеството или от друго подобно звено. Независимо от наименованието на звеното съдействието за осигуряване на качеството може да бъде оказано на екипа по проекта, на ръководството на изпълняващата организация, на клиента/спонсора, а също така и на други заинтересовани страни, които не са активно ангажирани с работа по проекта.

Осигуряването на качеството предоставя също така и рамка за непрекъснато усъвършенстване на процесите, която е итеративно средство за повишаване на качеството на всички процеси. Продължаващото усъвършенстване на процесите намалява брака и елиминира дейностите, които не добавят стойност. Това позволява на процесите да се изпълняват с повишени нива на ефективност и ефикасност.

Процесът на проверка на изискванията за качество и на резултатите от контролните измервания на качеството, е необходим, за да се гарантира, че се използват подходящи стандарти за качество и оперативни дефиниции.

4.5.4. Мерки за мониторинг, контрол, измерване на качеството на изпълнение на дейностите и на персонала във фазите на проекта

Всеки един от посочените четири компонента са част от прилаганата методика за управление на качеството, в основата на която стоят проследяването, контрола и одита по управление на качеството.

Качество на продукта

Методологията PMI въвежда концепцията „достатъчно добро качество“, която предлага по-ефективен подход от обичайния „повече е по-добре“, защото дефинира цел, която е или достижима или не, което от своя страна дава възможност за продължаване или прекратяване на проекта.

Моделите за достатъчно добро качество са изпробвани в проекти изпълнявани по PMI, като прилагането и изборът на модел зависи от спецификата на проекта.

На база опита при прилагането на PMI, за нуждите на проекта ще се прилагат следните два модела:

- Дефиниран процес – следва се процеса, за да може качеството да е резултат от самия процес, който се следва за създаване на системата. За целта считаме че процеса, който сме дефинирали е добър и резултатът от неговото прилагане ще бъде достатъчно добро качество на системата;
- Динамичен компромис – отчитат се много фактори. За целта се следва линията, че системата е достатъчно добра и няма критични проблеми, не критичните са незначителни и не са необходими повече усилия за подобряването ѝ.

Качество на процеса (Process Quality)

Качеството на процеса е съотносимост към степента, до която един процес, включващ измерители и критерии за качество, е бил предложен с цел създаването на софтуер.

Целта на измерването и оценката на качеството на проекта са следните:

- Управление на рентабилността на ресурсите;
- Управление и разрешаване на рисковете;
- Управление и осигуряване на бюджетите, плановите и качеството;
- Събиране на данни за подобряване на процеса.

Спазването на процесите ще доведе до високо качество на разработката, което от своя страна ще намали риска за създаване на продукт с лошо качество. Но и обратната ситуация не винаги е вярна, че създаването на продукт с високо качество, означава че процесът е бил спазен. Което от своя страна означава, че качеството на процеса се измерва не само със степента, до която процесът е бил спазен, но и със степента на постигнатото качество на продуктите, създадени от този процес.

За да се подпомогне оценката на качеството на процеса и качеството на продукта, PMI използва следните елементи:

- Задача (Task) - описание на задачата за изпълнение и необходимите стъпки за изпълнението;

- Насока (Guideline) - техники и практически съвети, полезни за изпълнението на задачата;
- Насоки за продукта и контролни списъци (work product guidelines and checklists) - информация за това как да се разработва, оценява и използва работната разработка;
- Шаблони (Templates) - модели или прототипи на работни продукти, даващи структура и насоки за съдържанието.

В тази точка сме описали **мерките за управление и контрол на качеството на работата на персонала**, които „Смарт Системс 2010“ ЕООД ще използва при реализацията на проекта. Ние смятаме, че мерките, които ще прилагаме при изпълнение на поръчката ще се гарантира качество на резултата.

Мярка 1: Измерване на качеството (Measuring Quality)

Измерването на качеството на продукта или на процеса, изисква събиране и анализ на информация. Анализът на събраната информация ще помогне за постигането на контрол върху проекта, в следствие на което проектът ще бъде управляван по възможно най-добрия начин. За целта се използват показатели, които измерват качеството на продукта и на процеса. Показателите изискват критерии за идентифициране и определяне на нивото на желаното качество, което следва да бъде достигнато. Нивото на качеството се договаря и може да се променя, поради което следва да се определи и утвърди от клиента на възможно най-ранния етап от жизнения цикъл на софтуерната разработка. Например в ранните итерации, появата на висок брой дефекти в приложението е допустима и приемлива, но дефекти в архитектурата са неприемливи и недопустими.

Критериите за приемане могат да се определят по няколко начина и могат да включват повече от един показател. В повечето случаи критериите за приемане включват следните показатели:

- Дефекти – брой идентифицирани дефекти, брой отстранени дефекти, брой неотстранени дефекти;
- Обхват на теста – % от кода и от потребителските случаи;
- Производителност – време за изпълнение на специфично действие (потребителски случай, операция, събитие). Този критерий се използва най-често при различните видове тестване – за производителност, за възстановяване на системата след срив и други тестове;
- Съответствие – показва степента до която всеки продукт, дейност, задача или стъпка трябва да отговаря на определени условия или стандарти и насоки;
- Приемливост или удовлетворение – използва се със субективни показатели като използваемост и др.

Мярка 2: Измерване на качеството на продукта

Измерването на качеството на продукта при софтуерните проекти се извършва чрез следните техники:

- Преглед или проиграване;
- Проверка/инспекция;
- При инспекцията се проверява доколко подготвеното отговаря на изискванията на клиента. В общия случай, този подход е приложим когато нещо конкретно се предава на клиента т.е. относително завършен модул или цялостен софтуерен продукт. В противовес, качеството на дадена услуга основно може да се осигури чрез одит. И това е обусловено от самата дефиниция на това що е (управляема) услуга – нематериален продукт, представляващ набор от конкретни техники, които в комбинация с предоставени ресурси са средство клиентът да получи стойност.
- Изпълнение на части или на целия софтуер.

Различните измерители се използват в зависимост от обхвата на целите на измерването на качеството.

Мярка 3: Измерване качеството на процеса

Измерването на качеството на процеса се постига чрез събиране на информация за измерители за знания и постижения:

- Степента на спазване на стандартите, насоките и изпълнението на приет процес;
- Статус/състояние на текущия процес на изпълнение на планираното изпълнение;
- Качество на създадените работни продукти.

Измерването на качеството на процеса се постига с помощта на някои от следните техники за измерване:

- Напредък – демонстриран потребителски случай или завършен етап;
- Вариация/отклонение – разлики между планирани и действителни графици, бюджет, изисквания към човешкия ресурс и др.;
- Измерители и показатели за качеството на продукта.

Мярка 4: Оценяване на качеството (Evaluating Quality)

През целия жизнен цикъл от гледна точка на управлението на качеството се извършва измерване и оценка на качеството на продукта и процеса.

Оценка на качеството се извършва при настъпило важно събитие, в края на фаза или когато продукта е създаден. Оценките, които може да се направят по време на жизнения цикъл са:

- Важни събития и оценки на статуса –всяка фаза и итерация имат за резултат пускането (вътрешно или външно) на продукт или подмножество на крайни продукти, които се разработват и по същото време се прави оценка на следните цели:
 - Демонстриране на постигането на изискванията;
 - Синхронизиране на очакванията;
 - Синхронизиране на свързаните работни продукти;
 - Идентифициране на рискове.

Основните важни събития настъпват в края на всяка от четирите фази - планиране, детайлизиране, изграждане и предаване, и потвърждават че целите са постигнати.

Оценката на статуса са периодичните оценки на постигнатия напредък през итерацията и/или фазата.

- Инспекции/одити и прегледи – насочени са към оценка на продукта и са ефективен метод за подобряване на качеството на процеса на разработка.

4.5.5. Приложимост на мерките за мониторинг и контрол към отделните фази на проекта

Таблица 5. Мерки за мониторинг и контрол във фазите на проекта

Фази/ Мерки за мониторинг и контрол	Анализ на данните и изискванията	Изготвяне на системен проект	Разработване на софтуерното решение	Тестване и внедряване	Обучение
Мярка 1	Не се прилага	Не се прилага	Прилага се	Не се прилага	Не се прилага
Мярка 2	Прилага се	Прилага се	Прилага се	Не се прилага	Не се прилага
Мярка 3	Прилага се	Прилага се	Прилага се	Прилага се	Прилага се
Мярка 4	Прилага се	Прилага се	Прилага се	Прилага се	Прилага се

4.5.6. Входни данни за осигуряване (измерване) на качеството

Осигуряването на качеството се осъществява на база следната входяща информация:

- План за управление на качеството, изготвен в предходния етап Планиране управлението на качеството;

- Информация за изпълнението на работата - Информацията за изпълнението на различни дейности по проекта се събира рутинно по време на неговото изпълнение. Данните за изпълнението, които подпомагат процеса по одитиране, включват, но не се ограничават до:
 - Технически показатели за ефективност;
 - Статус на резултатите от проекта;
 - Степен на изпълнение на графика;
 - Направени разходи.

4.5.7. Одити

Осигуряването (измерването) на качеството се осъществява чрез одити на качеството. Одитът на качеството е структуриран, независим преглед, който се провежда, за да бъде установено дали дейностите по проекта са в съответствие с политиките, процесите и процедурите на организацията и на проекта. Целите на одита на качеството са:

- Идентифициране на всички добри/най-добри практики, които са използвани;
- Идентифициране на всички пропуски/недостатъци;
- Споделяне на добрите практики, приложени при подобни проекти в организацията и/или бранша;
- Проактивно предлагане на съдействие по позитивен начин за усъвършенстване на изпълнението на процесите с цел повишаване на производителността на екипа;
- Подчертаване на приноса на всеки отделен одит в базата от знания с извлечени поуки.

Последващите усилия за коригиране на евентуалните несъвършенства следва да доведат до повишаване на удовлетвореността на клиента от продукта на проекта. Одитите на качеството могат да бъдат насрочвани по график или на случаен принцип и могат да бъдат провеждани от вътрешни или външни одитори.

Одитите на качеството могат да се използват за потвърждаване на прилагането на одобрените искания за промяна, включително коригиращи действия, отстраняване на дефекти и превантивни действия.

4.5.8. Натрупване на допълнително информация с цел анализ и бъдещи действия

По този начин се осигурява актуалност на средствата на организационния процес, в случая т. нар. Фирмена база от знания. Това представлява натрупана база данни от записи по предишно изпълнение на проекти в съответна област, в случая за предоставяне на услуги. Записите условно могат да бъдат разделени на две:

- Историческа информация – записи от предишни проекти – структура, планове, шаблони, отчети, рискове, оценки, използвани ресурси, кореспонденция.
- Изводи – конкретни стъпки, събития, които са важни за конкретния проект и може да се ползват за референция при изпълнението на подобни проекти или предоставянето на подобни услуги. Възможно е въз основа на изводите при изпълнението на даден договор по определена процедура да се извърши корекция или трайна промяна на някоя от използваните вътрешни процедури или политики от страна на Сمارт Системс .

Процесът по натрупване и ползване най-общо може да се опише със следната фигура:



Фигура 8 Изводи за конкретен проект

4.5.9. Извеждане на изводи и резултати от извършени одити

Процесът по Осигуряване (измерване) на качеството може да завърши с:

- Актуализирани организационни процесни активи - Елементите на организационните процесни активи, които могат да бъдат актуализирани, включват, но не се ограничават до стандартите за качество;
- Искания за промяна - Повишаването на качеството включва предприемането на действия за подобряване на ефективността и ефикасността на политиките, процесите и процедурите на изпълняващата организация. За да се осигури пълното обхващане на препоръчаните подобрения, се създават искания за промяна. Исканията за промяна могат да бъдат използвани и за предприемане на коригиращи или превантивни действия и отстраняване на дефекти;
- Актуализиран план за управление на качеството;
- Актуализирана документация на проекта - Документите по проекта, които могат да бъдат актуализирани, включват, но не се ограничават до:
 - Отчети от одити на качеството;

- Планове за обучение;
- Документация на процесите.

4.5.10. Анализ на процесите

Анализът на процесите следва стъпките, описани в плана за подобряване на процесите, така че да посочи необходимите подобрения. Анализът покрива и срещнатите проблеми, ограничения и дейности, които не добавят стойност, установени по време на изпълнението на процесите. Анализът включва анализ на първопричините - специфичен подход за идентифициране на проблем, разкриване на причините, които водят до него, и планиране на превантивни действия.

4.5.11. Инструменти и методи на контролиране на качеството

Контролирането на качеството е процесът на наблюдение и регистриране на резултатите от изпълнението на дейностите по качество, чрез които се оценява изпълнението и се правят препоръки за необходимите промени. Контролирането на качеството се извършва през целия проект. Стандартите за качество включват процесите по проекта и целите на проекта. Резултатите от проекта включват междинни резултати и резултати от управлението на проекта, като разходи и степен на изпълнение на графика. Контролирането на качеството често пъти се изпълнява от отдел за контролиране на качеството или от друго звено с подобно име. Дейностите по контрол на качеството идентифицират причините за ниско качество на процес или продукт и препоръчват и/или предприемат действия за тяхното отстраняване.

Методите и средствата, които се могат да се използват за контролиране на качеството, включват:

- **Причинно-следствени диаграми** - Причинно-следствените диаграми, наричани още „диаграми на Ишикава“ или „диаграми на рибената кост“, илюстрират как различни фактори биха могли да бъдат свързани с потенциални проблеми или ефекти. При продължително задаване на въпросите „защо?“ и „как?“ по някоя от линиите може да се достигне до вероятната първопричина за проблемите. Диаграмите от типа „защо – защо“ и „как – как“ могат да бъдат използвани при анализа на първопричините. Причинно-следствените диаграми се използват и при анализа на риска;
- **Контролни диаграми** - За този инструмент се събират и анализират подходящи данни, така че да се изясни статусът на качеството на процесите и продуктите по проекта. Контролните диаграми илюстрират начина на поведение на даден процес във времето и дали процесът се характеризира с отклонения, породени от специални причини, които да доведат до състояние на „контрол“ и „извън контрол“. Диаграмите дават графичен отговор на въпроса: „В рамките на приемливото ли се движат отклоненията на този процес?“. Линията от точки на данните върху контролната диаграма може да разкрие случайно променящи се стойности, резки скокове в процеса или постепенна тенденция към увеличаване

на отклоненията. Чрез наблюдение на изходите от процеса във времето контролната диаграма може да подпомогне преценката дали прилагането на промени в процеса води до желаните подобрения. Когато процесът е в рамките на приемливите граници, той е „под контрол“ и промени в него не са необходими. Обратно, когато процесът е извън приемливите граници, той трябва да бъде променен;

- **Процесни диаграми** - Процесните диаграми се използват при контролирането на качеството за установяване на проблемните стъпки на процесите и идентифициране на потенциалните възможности за усъвършенстване. Създаването на процесни диаграми се прилага и при анализа на риска;
- **Хистограма** - Хистограмата е вертикална лентова диаграма, която показва колко често се наблюдава конкретно състояние на променливата величина. Всеки стълб представлява отделен атрибут или характеристика на проблема/ситуацията. Височината на стълба представлява относителната честота на характеристиката. Този инструмент илюстрира най-често срещаните причини за проблеми в даден процес чрез броя и относителните височини на стълбовете.
- **Диаграма на Парето** - Диаграмата на Парето е специфичен тип хистограма, подредена съгласно честотата на случаите. Тя показва колко дефекти са причинени от различните типове или категории идентифицирани причини. Подреждането в класация се използва за насочване на коригиращите действия. Екипът по проекта трябва първо да обърне внимание на причините, които водят до най-голям брой дефекти;
- **Диаграма на тенденциите** - Сходна с контролната диаграма без посочени граници, диаграмата на тенденциите показва историята и линията на отклоненията. Диаграмата на тенденциите е линейна диаграма, която представя точките от данни, изобразени в хронологичен ред. Тези диаграми демонстрират тенденциите на процеса във времето, отклоненията във времето, влошаванията или подобренията. Анализът на тенденциите се извършва именно посредством тези диаграми и включва използването на математически методи за прогнозиране на бъдещите изходи въз основа на историческите данни. Анализът на тенденциите често се използва за наблюдение на:
 - Техническо изпълнение - Колко грешки и дефекти са установени и колко от тях са останали некоригирани;
 - Изпълнение на разходите и на графика - Какъв брой дейности през периода са извършени със значителни отклонения;
- **Инспекция** - Инспекцията представлява преглеждане на продукт от работата с цел установяване дали той съответства на документираните стандарти. Изходът от инспекцията обикновено включва данни от измервания, които могат да бъдат направени на произволно ниво. Например може да бъде инспектиран резултатът от единична дейност, а може да бъде инспектиран и крайният продукт на проекта. Инспекциите могат да бъдат наречени „прегледи“, „партньорски прегледи“;

„одити“ или „обхождания“. В някои области на приложение тези термини имат по-тясно и специфично значение. Инспекцията се използва и за одобряване на отстранени дефекти;

- **Преглед на одобрените искания за промяна** - Всички одобрени искания за промяна трябва да бъдат преглеждани, за да бъде проверено дали те са изпълнени във вида, в който са били одобрени.

4.5.12. Валидиране на резултати и мерки за контрол на промяна

Процесът по Контролиране на качеството може да завърши с:

- **Контролни измервания на качеството** - Контролните измервания на качеството са документираните резултати от дейностите по контрола на качеството във формата, определен при планиране на качеството;
- **Валидирани промени** - Всички променени или поправени обекти се инспектират и биват одобрявани или отхвърляни, преди да бъде изпратено известие за взетото решение. Отхвърлените единици могат да изискват преработка;
- **Валидирани резултати** - Една от целите на контролирането на качеството е установяването на коректността на резултатите по проекта. Резултатите от изпълнението на процесите по контролиране на качеството са валидирани изходи по проекта. Те представляват входяща информация за процеса по верифициране на обхвата с цел официалното им одобрение;
- **Актуализирани организационни процесни активи** - Елементите на организационните процесни активи, които могат да бъдат актуализирани, включват, но не се ограничават до:
 - Попълнени списъци за проверка на качеството - Когато се използват списъци за проверка, вече попълнените списъци стават част от отчетите по проекта;
 - Документация за извлечените поуки - Причините за отклонения, основанията за избраните коригиращи действия и други типове изводи, направени вследствие на контролирането на качеството, са документираны, така че да станат част от историческата база данни както на проекта, така и на изпълняващата организация. Поуките се документират през целия жизнен цикъл на проекта, но като задължителен минимум - при приключването на проекта;
 - Искания за промяна - Ако препоръчаните коригиращи или превантивни действия, или отстраняване на дефект изискват промени в плана за управление на проекта, трябва да бъде инициатирано искане за промяна;
 - Актуализиран план за управление на качеството;

- Актуализирана документация на проекта - Документите по проекта, които могат да бъдат актуализирани, включват, но не се ограничават до стандартите за качество.

Процесът на наблюдение и регистриране на резултатите от изпълнението на дейностите по качество, чрез които се оценява изпълнението и се правят препоръки за необходими промени.

В рамките на този процес, който много прилича на предишния процес по гарантиране на качеството, се извършва проследяване на конкретни резултати, за да се определи дали отговарят на зададените стандарти и/или очаквания. В случаи, че има нужда се набелязват начини за отстраняване на причините за незадоволителните резултати. Основната разлика между този и предния процес е, че предния процес се интересува как се извършва дадена дейност спрямо предварително фиксирани рамки, определени стандарти и т.н., а в този процес фокусът е върху изпълненото т.е. дали вече изработеното отговаря на поставените изисквания. За сравнение, спазването на срока по предаването на конкретни отчети е качествен показател за гарантиране на качеството, защото ще рефлектира върху процеса по подготовка на съответните отчети, докато проверката на съдържанието му може да върне отчетите за преработка или да ги допусне за предаване на клиента, което е процес по контрол на качеството.

Визуално процесът се представя със следната фигура:



Фигура 9 Контрол на качеството

И този процес се извършва в хода на изпълнение на дейностите от по целия проект и се ползва следния инструментариум:

- Тестване на софтуерен продукт;
- Мониторинг на качествени показатели и сравнението им с определени реперни стойности.

Сравняването на качествени показатели позволява да се провери съответния показател спрямо определени очаквани стойности. Тези очаквани стойности може да се фиксират въз основа на:

- Стойности фиксирани в заданието;

- Стойности на същите или подобни показатели в подобни проекти;
- Стойности, които са приети като реперни стойности според добри практики, средни стойности в дадения бранш или др. подобни източници;
- Стойности, определени въз основа на опита на „Смарт Системс 2010“ ЕООД. В конкретния случай, има два вида показатели, които ще се проследяват:
 - Показатели от спецификациите на системите, свързани с нефункционалните изисквания към системите. Те ще се проверяват с планираните системни тестове.
 - Показатели извлечени от плана на проекта, които ще се ползват за ефективното управление на проекта.
- Одит - Смарт Системс т ще организира провеждане на вътрешни одити, имащи за цел да проследят качеството на изпълнение на дейностите, сигурността и риска. Изпълнението на тези одити ще минава през следните стъпки:
 - Планиране на одит - В рамките му ще се подготви програма за провеждане на одита в съответствие на поставените цели, в която се отразяват:
 - Обхват (задачи и въпроси) на одита;
 - Време и срок за изпълнение;
 - Конкретен участник в одита от екипа по проекта;
 - Място и метод за проверка.
 - Провеждане на одит - Набавянето на необходимата информация за процесите, свързани с изпълнението на одита, се извършва под формата на провеждане на интервюта и/или преглед на наличната документация.

При откриване на несъответствие, то се обсъжда с представителя на одитираното звено, като се обсъждат и уточняват обективните свидетелства.

За всяко конкретно несъответствие, се предвиждат превантивни мерки или коригиращи действия и сроковете за изпълнението им.

- Приключване на одит - След приключване на одита, одитора предоставя информация за:
 - Резултатите и констатациите от одита;
 - Препоръки (ако има такива);
 - Несъответствие/я (ако има такива).
 - Управление на промените
 - Управление на риска

4.5.13. Приложение на подхода за управление на качеството във фазите на проекта

В рамките на този процес се изпълняват всички планирани и систематични действия в рамките на системата за качество, които дават увереност, че проектът ще отговаря на съответни стандарти, процедури. Процесът се изпълнява и извършва в хода на изпълнение на дейностите от проекта, обхващащ задълженията на Смарт Системс .

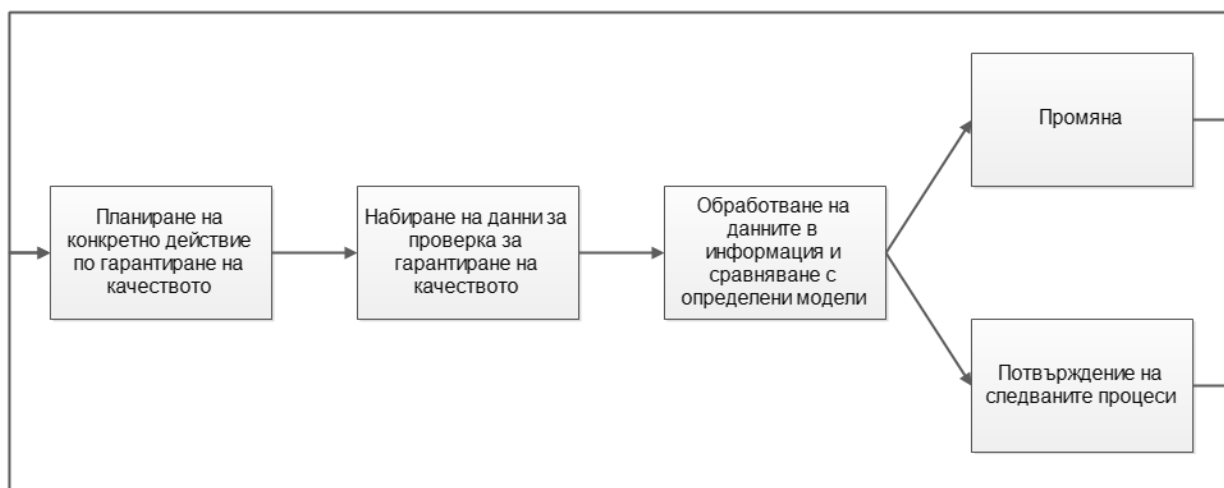
Или казано по-просто, в рамките на този процес ще се изпълняват планираните вече действия, с които се осигурява изискваното качество.

Дейностите извършвани в рамките на този и горния процес са от типа на т.нар. вградено качество т.е. предварително се изпълняват дадени процеси, стъпки, дейности около основния процес (по изпълнение на конкретна дейност при предоставяне на услугата), така че резултата от този процес да е с високо качество – възможно най-близко до очакванията на потребителя.

Важно е да се подчертае разликата между гарантиране на качеството и контрола му. При гарантирането се проверяват определени правила (въведени при планирането) въз основа на който ще се получи продукта на проекта с високо качество - в конкретната поръчка това са начина на проектиране на ИС със съответните функционалните и нефункционални изисквания към нея. Контролът на качеството се извършва, когато вече имаме изпълнено действие т.е. изследва се някакъв готов продукт на конкретна дейност.

При извършването на процеса по гарантиране на качеството, изходните резултати са както следва:

- Промяна – казано в най-общ смисъл. Може да е коригиращо действия, превантивна мярка, обновяване. В случаите, когато е свързано с промяна в официално фиксирани параметри по изпълнението на дейностите, то тогава тази промяната ще се реализира чрез въведената процедура за управление на промените.
- Потвърждение на следваните процеси. Проверката установява, че използваните към момента процеси са адекватни и не се изисква конкретно действие. Процесът може визуално да се представи по следния начин:



Фигура 10 Гарантиране на качеството

Отделните стъпки са както следва:

- Планиране на конкретно действие по гарантиране на качеството – има се предвид да се планира като график, определената дейност, която вече е избрана коя ще е и как ще се изпълнява в рамките на процеса по Планиране на качеството

- Набиране на данни за проверка за гарантиране на качеството – извършване на одит, подготовка на документ с изчерпателни тестови сценарии;
- Обработване на данните в информация и сравняване с определени модели – изпълнение на тестове и сравняване на получените резултати с желаните, които са следствие на изискванията към системите;
- Изходни процеси – така както са описани по горе.

За гарантирането на качеството от описаните по-горе различни видове инструментариум ще се ползва:

- Одит;
- Мониторинг на качествени показатели и сравнението им с определени реперни стойности;
- Поддържане на база данни от знания;
- Обучение;
- Управление на промените;
- Управление на риска.

Тези процеси взаимодействат помежду си и с останалите процеси в проекта. Всеки процес може да включва работата на едно или повече лица или групи в зависимост от изискванията на проекта. Всеки процес протича поне по веднъж за даден проект, като това се случва по време на една или повече от неговите фази. Независимо че процесите са представени като отделни компоненти с ясно дефинирани интерфейси, на практика е възможно те да се припокриват и да си взаимодействат.

Управлението на качеството в проекта засяга цялостното управление на проекта и на продукта, резултат от него. С оглед на това „Смарт Системс 2010“ ЕООД предлага в настоящата поръчка да бъде приложен комплексен подход за управление на качеството с прилагане на процесите за управление на качеството, дефинирани в методологията за управление на проекта.

Качеството на софтуерната разработка ще се проследява във всеки един момент от фазите на разработка, а именно: Планиране, Детайлизиране, Изграждане и Предаване.

Съгласно RUP, качеството при софтуерната разработка се дефинира като „характеристика на доказано постигнато създаване на продукт, който отговаря или надвишава договорните изисквания, измерени чрез договорени измерители и критерии и е създаден при спазването на договорения процес“.

При реализацията на проекта ще бъдат идентифицирани измерителите и критериите, с които да се потвърди постигането на качеството във всяка една от итерациите на проекта. Итерацията е верига на пълното развитие, чийто резултат е пускането на завършен продукт, част от крайния продукт, който се разработва и неговата функционалност се увеличава от итерация на итерация, за да се превърне в крайната система.

Всяка итерация преминава през всички етапи на софтуерната разработка, които са всъщност потоци от дейности.



Фигура 11 Модел на итерациите

Дейностите включват:

- Бизнес моделиране;
- Управление на изискванията;
- Анализ и дизайн;
- Реализация;
- Тестване;
- Внедряване (в последната итерация).

Крайният продукт на една итерация представлява документирана част от завършената система.

В следствие от прилагането на този процес се гарантира, че постигането на качество се управлява и гарантира непрекъснато, през целия жизнен цикъл на проекта.

Управлението на качеството се базира на следните четири компонента:

- Качеството на продукта (Product Quality);
- Качеството на процеса (Process Quality);
- Измерване на качеството (Measuring Quality);
- Оценяване на качеството (Evaluating Quality).

4.5.14. Използване на методи и подходи за вземане на решения, основани на действителни/реални факти и информация (Factual approach to decision making)

При този принцип съществува взаимовръзка между управленската наука и практика. Известно е, че едно решение ще бъде добро, когато е обосновано, притежава ясна формулировка и осигурява гъвкавост на неговото изпълнение, за изпълнението му организацията е в състояние да осигури необходимите ресурси, изразява по най-добрия начин интересите на лицата и организациите, които имат отношение към него и др. Информацията е от съществено значение за постигането на всички тези изисквания. Поради това този принцип акцентира именно върху нея и то върху онази, която е достатъчно точна и не е основана на прогнози и предположения. Основните съображения за това са свързани с целите на този принцип, а именно прилагайки го да се осигурят:

- достатъчна по количество и качество информация за процесите на взимане на решения;
- по-добра ефективност на процесите на взимане на решения, когато за целта се използва вярна, достатъчна и навременна информация;
- възможности за взимането на качествени решения в условията на зараждащи се или настъпили промени.

Предвидените в принципа действия, които трябва да бъдат извършени, за да се осигури постигането на посочените условия са:

- гарантиране на навременното, точното и пълното получаване и разпределение на информацията в екипа по изпълнение на проекта;
- използването на научни методи за анализ на необходимата информация;
- взимане на решения само на основата на резултатите от извършения анализ на фактическите резултати и постигане на баланс между тях, опита и интуицията на ръководството.

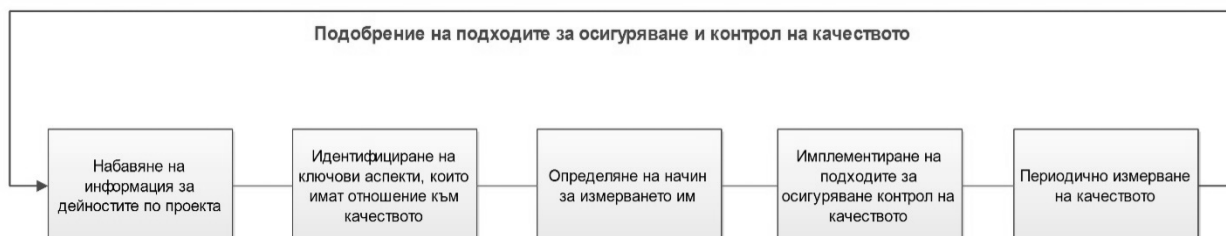
Планиране управлението на качеството

Процесът на определяне на изискванията за качество и/или приложимите стандарти към проекта и продукта, както и документиране как проектът ще покаже съответствие с тях.

Планирането на качеството е процесът на определяне на изискванията за качество и/или приложимите стандарти към проекта и продукта, както и документиране как проектът ще покаже съответствие с тях. Планирането на качеството се извършва по фази и дейности съобразно спецификата на проекта и обхваща:

- Дейностите и етапите в проекта;
- Изисквания към входните и изходните данни;
- Изисквания към задълженията на участниците и отговорните лица;
- Утвърждаване и проверка на процеси и продукти;

- Анализи на процесите;
- Оценка на риска;
- Коригиращи и превантивни действия -
- Управление на измененията на процесите и продуктите;
- Необходимите записи като обективно доказателство за извършените процеси.



Фигура 12 Подобрение на подходите за осигуряване и контрол на качеството

4.5.15. Подход за изготвяне на план за управление на качеството

На база наличните входни данни се изготвя Планът за управление на качеството (ПУК). Планът за управление на качеството описва начина, по който екипът за управление на проекта ще приложи политиката по качеството.

Планът за управление на качеството предоставя входни данни за цялостния план за управление на проекта и включва подходите за контролиране на качеството по отношение на процесите и продукта, осигуряване на качеството и продължаващо усъвършенстване на процесите по проекта.

Планът за управление на качеството включва показатели за качество. Показателят за качество е оперативна дефиниция, която със специфична терминология, описва даден атрибут на проекта или продукта и начина, по който процесът за контролиране на качеството ще го измери. Измерената стойност е действителната стойност. Допустимите стойности определят позволените отклонения от показателите. Показателите за качество се използват в процесите по осигуряване и контролиране на качеството. Примерите за показатели за качество включват степен на придържане към графика, контролиране на бюджета, честота на дефектите, дял на дефектния продукт, наличност, надеждност и покритие на тестовите.

Планът за управление на качеството може да включва и списъци за проверка на качеството. Списъкът за проверка на качеството е структуриран инструмент, обикновено специфичен за компонента, използван за удостоверяване, че даден набор от задължителни стъпки е изпълнен. Списъците за проверка могат да бъдат прости или сложни в зависимост от изискванията и практиките по проекта. Много организации притежават свои стандартизирани списъци за проверка, които се прилагат за осигуряване на съгласуваност при често изпълнявани задачи. За някои области на приложение могат да се употребяват стандартизирани списъци за проверка, разработени от професионални асоциации или търговски доставчици на услуги. Списъците за проверка на качеството се използват в процеса по контролиране на качеството.

Цел на Плана по качеството е определяне и документиране на дейностите, свързани с качеството, които ще се прилагат в хода на реализацията на проекта, с цел:

- да се осигури съответствие на междинния или крайния продукт с изискванията на проекта;
- да се осигурят ефективни комуникации между всички участници в процеса за реализация на проекта;
- да се осигури адекватен контрол на всички промени в план-графиците, спецификациите, и пр.;
- да се осигури унификация между участниците в проекта по отношение на стандартите, процедурите и методите, които се прилагат за постигане целите на проекта;
- да се приложат процедури за осигуряване доставката на продукт, съответстващ на изискванията на договора и отговарящ на стандартите и критериите за качество и сроковете за доставка.

Планът по качеството описва процесите за управление на взаимовръзките между Възложител и Изпълнител в следните направления:

- въвеждане на разпознаваеми за двете страни термини и понятия, съкращения и акроними и поддържане на терминологичен речник, които се ползват и еднозначно разбират от всички участници в жизнения цикъл на проекта;
- определяне на цели и критерии по качеството;
- комуникации – участници, форми;
- план-графици на проекта;
- записи по качеството, изисквани документи – формат за изготвяне и конвенции за именуване, форма за съхранение, момент на изготвяне, схема за разпространение и съгласуване, правила за утвърждаване;
- контрол на развитието – наблюдение и оценки;
- преглед на продукта – преглед на дизайна, дейности за верификация и валидация (функционални тестове),
- управление на промени – процедура;
- ескалация и решаване на проблеми - процедура;
- контрол на постигнатите резултати в контролни точки от план-графиците;
- управление на риска – процедура;
- контрол на качеството – одити, прегледи, други методи;
- сигурност и защита – приложими методи относно различните елементи и процедури за реализация, конфиденциалност, план по сигурността;

- доставка на продукта документи, процедура за приемане;
- обучение на различни категории персонал за експлоатация на продукта;
- условия за инсталация и инсталация;
- миграция на данни;
- предоставяна експлоатационна документация за приложния софтуер и за базовия софтуер;
- гаранционно поддържане;
- процедура за превантивни и коригиращи действия за отстраняване на несъответствия в процеса на реализацията на проекта;
- идентификация и управление на риска;
- приложими стандарти, процедури и методи;
- роли и отговорности;
- приложими форми за документиране на записи по качеството.

4.6. График за изпълнение на проекта

График на всички дейности и под дейности в проекта е включена в „Приложение 2.2 към Техническото предложение – Детайлен график по дейности и под-дейности от проекта“

Предлагания график е в съответствие с всички изисквания по отношение на сроковете за изпълнение на дейностите залегнали в Техническата спецификация, както и с посочения срок за изпълнение на проекта (сроковете са изчислени спрямо датата на подписване на договор, а като примерна дата за начало на проекта е приет **02.01.2020 г.**), както следва:

Дейност	Начална Дата	Крайна Дата
Встъпителен доклад		14.01.2020
Дейност 1: Изготвяне на системен проект	07.01.2020	03.04.2020
Системен проект		02.04.2020
Междинен доклад за Дейност 1		03.04.2020
Дейност 2: Развитие на информационна система „Външна търговия“ за миграция на платформата на модул „Микроданни“ чрез използването на съвременни високотехнологични ИТ решения	03.04.2020	11.09.2020

Междинен доклад за Дейност 2		11.09.2020
Дейност 3. Развитие на информационна система „Външна търговия“ за добавяне на нови функционалности и промяна на съществуващите.	04.09.2020	22.10.2020
Междинен доклад за Дейност 3		22.10.2020
Дейност 4. Тестване на системата	12.08.2020	25.11.2020
Междинен доклад за Дейност 4		25.11.2020
Дейност 5. Обучение	14.10.2020	15.12.2020
Междинен доклад за Дейност 5		15.12.2020
Дейност 6. Внедряване	08.12.2020	17.12.2020
Окончателен доклад		30.12.2020
Предлаган срок за изпълнение на проекта		
Гаранционна поддръжка за 36 месеца и отстраняване на грешки във функционирането на софтуера, след приемане в експлоатация на системата По време на гаранционната поддръжка Изпълнителят ще осигури минимум 5 свои специалисти в продължение на минимум 30 работни дни		

4.7. Управление на риска

През времето за изпълнение на проекта ще бъдат следени посочените по-долу рискове. Ще се оценява тяхното влияние, ще се анализира ситуацията и ще се идентифицират и управляват (евентуално) нови рискове.

- Промяна в нормативната уредба, водеща до промяна на ключови компоненти на решението – предмет на разработка на настоящата обществена поръчка;
- Недобра комуникация между екипите на Възложителя и Сمارт Системс по време на аналитичните етапи на проекта;
- Ненавременно изпълнение на всяко от задълженията от страна на Смарт Системс ;
- Неправилно и неефективно разпределяне на ресурсите и отговорностите при изпълнението на договора;
- Забавяне при изпълнение на проектните дейности, опасност от неспазване на срока за изпълнение на настоящата поръчка;
- Грешки при разработване на функционалностите на системата; Недостатъчна яснота по правната рамка и/или променяща се правна рамка по време на изпълнение на проекта;
- Липса на задълбоченост при изследването и описанието на бизнес процесите и данните;
- Неинформирание на Възложителя за всички потенциални проблеми, които биха могли да възникнат в хода на изпълнение на дейностите;
- Риск за администриране на системата след изтичане на периода на гаранционна поддръжка.

Понятието „риск” се отнася до несигурни бъдещи събития или условия, които, ако възникнат, биха имали ефект върху поне една от целите на проекта - обхват, план-график, бюджет и качество.

Доколкото действителната стойност на този ефект може да бъде в положителна насока (рискове представляващи възможности) или отрицателна насока (рискове представляващи заплаха), управлението на риска се фокусира върху увеличаване на потенциалните ползи и намаляване на потенциалните щети, които могат да възникнат от бъдещите неопределени действия. Тези щети може да се състоят от претърпени загуби или от пропуснати ползи.

Управлението на риска е един от най-критичните аспекти на всеки един ИТ проект, отчитайки емпирично доказания факт, че неуправляваните рискове могат сериозно да застрашат постигането на целите на проекта, а от там и цялостния му успех. Това налага изключително сериозно и професионално отношение към избора на най-подходяща методология за управление на риска за конкретния проект, а така също и към нейното прилагане през целия жизнен цикъл на проекта.

Управлението на риска е комплекс от дейности, чиято цел е да увеличат вероятността и/или последствията от позитивните събития и съответно да се намалят вероятността и/или последствията от негативните събития, което в крайна сметка е насочено към увеличаване на възможностите за успех на проекта.

Съгласно методологията RUP, управлението на риска се състои от следните дейности:

- **Планиране** – дефинира се как ще се провеждат дейностите по управление на риска;
- **Идентификация** – определяне кои рискове могат да повлияят на проекта и документиране на техните характеристики;
- **Анализ**
 - **Качествен анализ (Приоритизация на рисковете)** – приоритизиране на рисковете за бъдещ анализ или действия чрез оценка и комбиниране на тяхната вероятност от възникване и въздействието при такова възникване, включващо и **класифициране/категоризиране** на рисковете по дефинирани при планирането критерии;
 - **Количествен анализ** – числен анализ на ефекта от идентифицираните рискове към целите на проекта;
- **Планиране на реакцията** – разработване на варианти за действия за увеличаване на положителните влияния и намаляване на заплахите;
- **Наблюдение и контрол** – прилагане на планове за реакция, проследяване на идентифицираните рискове, идентифициране на нови рискове и оценка на ефективността на управлението на риска.

Основните принципи на управлението на риска са следните:

- ✓ Добавяне на стойност – ефекта от намаляване на риска трябва да надхвърлят цената на реакция;
- ✓ Неразделна част е от всички процеси в организацията;
- ✓ Част е от вземането на решения;
- ✓ Изцяло адресира само несигурности и допускания – събития, които е сигурно, че ще се случат, не са риск;
- ✓ Управлението на риска е систематизирано и структурирано;
- ✓ Основава се на цялата налична информация;
- ✓ Взема се предвид човешкия фактор;
- ✓ Извършва се динамично и с откликване на промените.

По-надолу ще бъде разписани детайлно дейностите в обхвата на предлаганата Методология за управление на риска в проекта.

4.7.1. Планиране на управлението на риска

Внимателното и пълно планиране увеличава вероятността от успех на останалите дейности и процеси от управлението на риска.

Планирането се извършва на срещи на екипа, включващи ръководителя на проекта, ръководителя на екипа за изпълнение, избрани членове на екипа и представители на клиента (Възложителя), които имат отношение към риска и други, ако са необходими.

Планът за управление на риска описва как ще бъде структурирано и изпълнявано това управление и може да включва:

- **Методология** – подход, инструменти и източници на данни за управлението на риска;
- **Бюджет** – когато се предвижда обособено финансиране за управление на риска;
- **График** – определя кога и колко често ще се извършват дейности по управление на риска, разпределени във времето по фазите и итерациите на проекта;
- **Роли и отговорности** – дефинира ясно и детайлно ролите и отговорностите на участниците в екипа за изпълнение на проекта по отношение на дейностите по управление на риска;
- **Критерии за категоризиране/класифициране на рисковете** – категоризирането/класифицирането представлява структуриране на категориите рискове с цел подпомагане на процеса за систематично идентифициране на рисковете. В тази част на Плана за управление на риска се дефинират подходящите за проекта критерии за категоризиране/класифициране на рисковете – например по дейностите на проекта, по степента на влияние върху целите на проекта и/или по най-срещания подход за категоризиране - по източниците на рисковете (потребители, изисквания, технологии, планиране и контрол, екип за изпълнение и организационна среда);
- **Дефиниране на метрики** – общо определяне на нивата на вероятност и влияние на риска, съобразени с проекта.

4.7.2. Идентифициране на рисковете

Това е процес на определяне кои рискове могат да повлияят на проекта и документиране на техните характеристики. В идентифицирането на рисковете е добре да бъдат окуражавани да вземат участие всички участници в проекта, но е задължително за ръководителя на проекта, отговорника по управление на риска, ако има назначен такъв (в нашия случай това е Експерта управление на качеството и тестване на информационни системи) и представители на Клиента (Възложителя), които имат отношение към риска.

Идентифицирането на рисковете е итеративен процес, защото нови рискове могат да възникнат или част от старите да отпаднат.

Съществуват различни техники за идентификация на рисковете, но всички те се основават на експертните познания.

Такива техники могат да бъдат:

- **Брейнсторминг** – целта е да се получи подробен списък на рисковете;

- Интервюта – интервюират се опитни участници в проекта и експерти в областта на проекта;
- Анализ на първопричините – идентифициране на проблемите чрез анализиране на първопричините за тях;
- Анализ на списъци – анализира се историческа информация и познание, придобити от предишни подобни проекти. Възможно е да се използват и външни източници;
- SWOT анализ – изследване на проекта от гледна точка на силните и слабите страни, възможностите и заплахите, за разширяване погледа към рисковете;
- Експертни оценка – рисковете могат да бъдат идентифицирани директно от експерти с релевантен опит в подобни проекти.
- Резултатът от идентификацията на рисковете е **Регистър на рисковете**, който се състои от:
 - **Списък на идентифицираните рискове** – описани са в колкото се може повече подробности;
 - **Списък на потенциалните реакции** – когато могат да бъдат идентифицирани още с откриването на риска или добавени впоследствие при изпълнение на дейността „Планиране на реакцията“.

4.7.3. Анализ

4.7.3.1. Качествен анализ (Приоритизация на рисковете)

Това е процес на приоритизация на рисковете за бъдещ анализ или действия чрез оценка и комбиниране на вероятността за тяхното възникване и влияние. Организацията може да подобри ефективността на проекта като се фокусира върху рисковете с по-висок приоритет.

Качественият анализ е бързо и ефективно средство за установяване приоритетите при планиране на реакциите на рисковете и е база за провеждане на количествения анализ.

За провеждане на качествения анализ се използва последователност от техники.

- Оценка на вероятността от настъпване - изследва се вероятността за възникване на всеки специфичен риск. Обикновено вероятността се изразява с процент в десетици или стотици в скалата от 0 до 1;
- Оценка на степента на значимост – изследва се ефекта (влиянието) в случай на реализиране на всеки един специфичен риск, т.е. отклонението от планираните графици, усилия и разходи, ако рискът действително се реализира;
- Оценка на Рисковата експозиция - матрица на вероятността от настъпване и степента на значимост (влиянието) – приоритизиране на рисковете, базирано на направената оценка. Получава се като произведение от

степеня на значимост и вероятността от настъпване. Обикновено се прави чрез матрица, както е показано на следващата фигура, като вероятността, степеня на значимост и рисковата експозиция се изразяват с процент или с точност до десетици или стотици в скалата от 0 до 1. Когато не е възможно точното остойностяване на вероятността и влиянието, а от там и на рисковата експозиция, се прилагат агрегиращи стойности като например „нисък, „среден“ и „висок“ по RUP. По време на етапа по планиране на риска се договаря значението на тези стойности.

Таблица 6 - Оценка на Рисковата експозиция - матрица на вероятността

Вероятност				
0,90	0,09	0,27	0,54	0,81
0,70	0,07	0,21	0,42	0,63
0,50	0,05	0,15	0,30	0,45
0,30	0,03	0,09	0,18	0,27
0,10	0,01	0,03	0,06	0,09
Степен на значимост (Влияние)	0,10	0,30	0,60	0,90

- Оценка на качеството на данните – оценява се степеня полезност на данните относно рисковете;
- Категоризация/класифициране на рисковете – идентифицираните рискове се категоризират/класифицират по критериите, дефинирани в Плана за управление на риска. Това е изключително важна дейност предвид на интерактивността на процеса по управление на риска – дефинирането на категории подпомага ефективното идентифициране на рисковете, а класифицирането на идентифицираните рискове подпомага дефинирането на адекватна реакция за противодействие на съответния риск.
- Определяне на индикатори за реализация/проявление на рисковете – за всеки риск от списъка се идентифицира подлежащо на измерване състояние, настъпването на което означава, че рискът действително се е реализирал/проявил;
- Оценка на критичността (спешността) – рисковете, които изискват реакция в по-кратки срокове се оценяват като по спешни за третиране;
- Експертна оценка – прилага се за оценка на вероятността и влиянието и поставяне на всеки риск на правилното място в матрицата на вероятността и влиянието.

В резултат на качествения анализ могат да се създадат следните документи:

- Промени в Регистъра на рисковете;
- Относително класиране или приоритизиране на рисковете;
- Групиране на рисковете по категории;
- Причини за рискове или области от проекта, които изискват повече внимание;
- Списък на рисковете, които изискват незабавно или бързо третиране.

4.7.3.2. Количествен анализ

Количественият анализ се извършва върху рисковете, които са определени като приоритетни в резултат на качествения анализ. Извършва се оценка на ефекта от тези рискови събития.

Количественият анализ в общи линии следва качествения анализ, но понякога може да не се изисква, за да се разработят реакциите на рисковете.

Техники за количествен анализ

Прилагат се най-разнообразни техники за количествен анализ, като някои от тях могат да бъдат следните:

- Интервюта – провеждат се с подходящите експерти като обикновено се оценяват три нива на информация: оптимистично (ниско), песимистично (високо) и най-вероятно;
- Разпределение на вероятността – използват се моделиращи и симулационни инструменти, които представят несигурността на стойностите на променливи величини като продължителност на дейности и цена на компоненти;
- Метод на очакваните стойности – математическа и статистическа концепция за изчисляване на възможни стойности при различни сценарии;
- Експертна оценка – изисква се за идентифициране на потенциалните разходи, оценка на вероятности и дефиниране на входните данни за използваните инструменти.

В резултат на количествения анализ могат да се създадат следните документи:

- Промени в Регистъра на рисковете;
- Вероятностни анализи;
- Вероятност за постигане на заложените срокове и бюджет;
- Приоритизиран списък на количествен ценени рискове.

4.7.3.3. Планиране на реакцията

Целите на планирането на реакцията са да се ограничат рисковете представляващи заплахата (негативните рискове) и да се максимизират рисковете даващи възможности (положителни рискове). Това е процес, който следва качествения анализ и ако е приложен количествения анализ. Включва и идентификацията на конкретна личност (собственик на риска), който е отговорен за третиране на риска.

Планираната реакция трябва да е адекватна на важността на риска, ценово ефективна и реалистична в контекста на проекта.

Техники за планиране на реакцията

Познати са различни стратегии за реакция на риска. За всеки риск трябва да се избере ефективната стратегия или съвкупност от стратегии. По-долу са представени най-често използваните стратегии за третиране на негативни рискове и заплахи.

- Избягване

Състои се в неизпълнение на дейност, която може да носи риск. Такъв пример е да не се лети с самолет на авиокомпания за избягване на риска от отвлечане. Избягването може би изглежда отговор на всички рискове, но това означава загуба на потенциалните ползи, които тази дейност би донесла, ако се приеме риска. Така, в горния пример, би ни се наложило да пътуваме много по-дълго с друг транспорт, избягвайки риска.

Част от избягването на риска е предпазване от опасности и се отнася до настъпили спешни събития. Първата и най-ефективна стъпка е елиминирането на опасността, но ако отнема твърде дълго време, твърде е скъпо или по друг начин непрактично, следващата стъпка е ограничаването на риска и неговото въздействие.

- **Намаляване/ограничаване**

Намаляването на риска (или ограничаване или оптимизиране) се състои в предприемане на действия, които водят до намаляване обема на загубите от щетата или намаляване вероятността за нейното настъпване. В някои случаи инструментите за намаляване на риска може да доведат до други рискове или да са толкова ресурсоемки, че да не оправдават тяхното използване. В примера със самолета решението да се пътува с автомобил би довело до поява на риск от задръстване, например, а решението за пътуване със собствен самолет по очевидни причини не е широко приемливо.

Разбирайки, че рискът може да е положителен или отрицателен, оптимизирането на риска означава намирането на баланс между негативите, които риска носи, в сравнение с ползите, които произтичат от изпълнението на дейността, както и между степента на намаляване на риска и необходимите усилия за това.

Модерните методологии за разработка на софтуер като RUP намаляват риска чрез разработване и инкрементално представяне на прототипи. Старите методологии страдаха от факта, че доставят софтуера на финалната фаза и всички възникнали на по-ранен етап проблеми означават скъпа преработка и често обричат проекта на провал.

- **Споделяне**

Краткото описание на споделянето е „споделяне с друга страна на тежестта от загубата или пропуснатата полза от риска, както и усилията, необходими за прилагане на мерките за намаляването му”.

- **Трансфер**

Трансфер предимно на финансовите рискове към трето лице, най-вече застраховател или доставчик. Тази реакция е припозната най-вече от методологията Prince2.

- **Приемане**

Представява приемане на загубите или пропуснатите ползи от риска, когато събитието настъпи. Само застраховането (заделяне на средства за компенсация на евентуални бъдещи загуби) попада в тази категория. Приемането на риска е съществена стратегия за малки рискове, където цената от подсигуриране срещу риска с течение на времето ще надмине общия сбор на очакваните загуби.

Всеки риск, който не е избегнат или прехвърлен към трета страна по подразбиране е приет. Това включва рискове, които са толкова големи или катастрофални, че не могат да бъдат подсигурени или цената на осигуровката е невъзможна. Форсмажорните обстоятелства (война, природно бедствие и др.) са примери за такъв риск.

Друга причина за приемане на риска е вероятността за настъпване на събитие, причиняващо твърде голяма загуба, е твърде малка или необходимите ресурси за осигуряването са такива, че да пречат твърде много на целите на организацията.

- Резервираност (Fallback)

Това е реакция, развита добре в методологията Prince2. Състои се в предварително оставяне на резерви (от време, ресурси, действия), които да се използват за противодействие на риска в случай на проявление.

В резултат на планиране на реакцията могат да се създадат следните документи:

- Промени в Регистъра на рисковете;
- Решения за сключване на договори, относими към рисковете;
- Промени в плана за управление на проекта – график, бюджет и др.;
- Промени в техническата документация.

4.7.3.4. Наблюдение и контрол

Планираните реакции на рисковете, включени в Регистъра на рисковете се изпълняват през целия жизнен цикъл на проекта, но трябва да се извършва и постоянно наблюдение за нови, променящи се или вече остарели рискове.

Използват се следните техники за наблюдение и контрол:

- Преоценка на риска – много често в резултат на наблюдението се идентифицират нови рискове, преоценяват се текущи рискове или се затварят остарели и невалидни вече;
- Одити на рисковете – изследва се и се документира ефективността на реакция на рисковете относно тяхното третиране и основните първопричини за тях, както и се оценява процесът по управление на рисковете;
- Анализ на отклонения и тенденции – сравняват се постигнатите резултати с планираните и се прави преглед на тенденции в изпълнението на проекта;
- Анализ на резервите – сравняване на остатъка на заделените за непредвидени обстоятелства резерви и остатъчните стойности на рисковете, за да се прецени адекватността на тези резерви;
- Оперативки – управлението на риска е толкова по-лесно, колкото по често се прилага. Честите дискусии относно рисковете увеличават вероятността хората да идентифицират рискове и възможности;

В резултат на наблюдението и контрола могат да се създадат следните документи:

- Промени в Регистъра на рисковете;
- Промени в Плана за управление на проекта;
- Искания за промяна (change requests);

- Промени в документацията на проекта.

В заключение е важно да се уточни, че процеса на управление на риска е итеративен процес (и затова най-добре кореспондира с методологията RUP) – изложените дейности по управление на риска се изпълняват циклично през целия жизнен цикъл на проекта. Първоначално идентифицираните, класифицираните, приоритизирани и с планирана реакция рискове подлежат на дейността „Наблюдение и контрол“, през която могат да се идентифицират и нови рискове за които отново следва да изпълнят итеративно дейностите по класифициране, приоритизация и планиране на реакцията, а за част от вече идентифицираните рискове може да се наложи нова приоритизация, предвид на настъпилите нови обстоятелства.

4.7.4. Прилагане на Методологията за управление на риска на проекта

След като изложихме теоретичната рамка на Методологията за управление на риска в проекта в тази точка ще представим нейното практическо прилагане в проекта.

Практическото прилагане на методологията за управление на риска в проекта се базира на подхода на RUP за описване на процеса по изграждане на информационни системи в две измерения:

- ✓ **статично измерение** – описва дейностите по управление на риска по същество и детайлизира ролите и отговорностите участниците в екипа в процеса по управление на риска.
- ✓ **динамично (времево) измерение** – представя изпълнението на описаните в предложената методология дейности по управление на риска през целия жизнения цикъл на проекта (по фази и итерации).

Тъй като тези две измерения са неразделна част от една методология, чиято цел е да доведе ИТ проекта до успешно изпълнение, то нашият подход за прилагане на предлаганата Методология за управление на риска се базира на интегрираното и цялостно прилагане на тези две измерения. Това е в унисон и със споменатия вече силно изразен итеративен характер на процеса по управление на риска.

4.7.4.1. Планиране на управлението на риска

Считаме изпълнението на тази дейност и по-конкретно разработването на **План за управление на риска** за основополагащ и критичен фактор за успешното управление на риска в проекта. Затова предлагаме **Плана за управление на риска** да бъде добавен като допълнителен документ в Документацията по управление на проекта.

4.7.4.2. Статично измерение на дейността „Планиране на управлението на риска“

Тази дейност по същество включва **разработването и актуализирането на Плана за управление на риска**, в съответствие със структурата, показана в предложената Методология за управление на риска:

- **Методология** – ще бъде изложена предложената Методология за управление на риска в проекта;
- **Бюджет** – предвид на спецификата на обществените поръчки не се предвижда обособено финансиране за управление на риска и тези дейности ще се финансират

в рамките на общия бюджет на проекта;

- **График** – ще обобщава динамичното измерение за всички дейности по управление на риска в един общ график, който ще определя кога и колко често ще се извършват дейности по управление на риска, разпределени във времето по фазите и итерациите на проекта;
- **Роли и отговорности** – ще дефинира ясно и детайлно ролите и отговорностите на участниците в екипа за изпълнение на проекта в процеса по управление на риска, който ще бъдат изложени в статичното измерение на всяка една от дейностите по управление на риска;
- **Критерии за категоризиране/класифициране на рисковете** – отчитайки спецификите на проекта и добрите практики в тази област, предлагаме рисковете по проекта да се категоризират/класифицират по следните два критерия:
 - **по основните дейности на проекта** – общи рискове и рискове по двете основни дейности;
 - **по източниците на рисковете** – добрата практика е по този критерий рисковете да се категоризират в 6 направления: потребител/възложител, обхват и изисквания, технологии, планиране и контрол, екип за изпълнение и организационна среда;
- **Дефиниране на метрики** – по отношение на Рисквата експозиция следва да се определи скала за степента на критичност за целите на реакцията:

Таблица 7 – Рисква експозиция и степен на критичност

Рисква експозиция	Степен на критичност
< 0,1	много ниска
0,1 - 0,3	ниска
0,3 – 0,5	умерена
0,5 – 0,8	висока
0,8 >	много висока

4.7.4.3. Динамично измерение на дейността „Планиране на управлението на риска“

Планът за управление на риска ще бъде разработен в началото на изпълнението на проекта, като ще бъде прилаган през целия жизнен цикъл на проекта и при необходимост ще бъде актуализиран в края на всеки един от етапите, дефинирани в 5.4. от Техническата спецификация.

4.7.4.4. Идентифициране на рисковете

Тази дейност е също изключително критична за процеса на управление на риска, тъй като има за цел да идентифицира обекта на това управление - несигурните бъдещи

събития или условия, наречени още рискове, които, ако възникнат биха имали ефект върху целите на проекта.

✓ **Статично измерение на дейността „Идентифициране на рисковете“**

Идентификацията на рисковете е итеративен процес, в резултат на който се създава **Регистъра на рисковете**, включващ:

- **Списък на идентифицираните рискове** – описани са в колкото се може повече подробности;
- **Списък на потенциалните реакции** – когато могат да бъдат идентифицирани още с откриването на риска или добавени впоследствие при изпълнение на дейността „Планиране на реакцията“.

Предвид на важността и критичността на тази дейност за ефективното управление на риска **предлагаме работна версия на Регистъра на рисковете**, посочени от Възложителя в Техническата спецификация и предложим първоначално виждане за **потенциални реакции** за всички рискове.

Таблица 8 - Работна версия на Регистъра на рисковете

№	Идентифицирани рискове	Очаквано въздействие	Мерки за противодействие
1.	Рискове идентифицирани от Възложителя		
1.1.	Промяна в нормативната уредба, водеща до промяна на ключови компоненти на решението – предмет на разработка на настоящата обществена поръчка	Промяна на ключови компоненти на решението – предмет на разработка на настоящата обществена поръчка, което може да доведе до промяна на архитектура и срокове на изпълнение на поръчката.	Своевременно осигуряване на информация за предстоящи промени в нормативната уредба, за да може резултатите да бъдат съобразени с тях. Гъвкавост на разработките с възможност за параметризация и вариативност.
1.2.	Недобра комуникация между екипите на Възложителя и Сمارт Системс по време на аналитичните етапи на проекта	Забава на изпълнението на проекта. Неотразяване на някои процеси, които в следствие да не намерят място в самата Система и са важни за работата на Възложителя	Мерки за противодействие както следва: <ul style="list-style-type: none"> • Ръководителят на екипа, подпомаган от Тестовия инженер. Да създадат организацията за изпълнение на всеки етап от проекта, която задължително включва възлагане,

			<p>следене, координация и съгласуване на изпълнението на дейностите през етапа и осъществяване на непрекъсната комуникация с Възложителя;</p> <ul style="list-style-type: none"> • Регулярност на срещите на управленското ниво на проекта от страна на Сمارт Системс и Възложителя, която включва: • Седмични срещи за бърз преглед на прогреса по изпълнение на проекта и синхронизация на дейностите между всички членове на екипа за изпълнение на проекта. <p>Месечни срещи за детайлен преглед на прогреса по изпълнение на проекта, обсъждане на ресурсното обезпечаване на проекта, анализ на възникналите проблеми и задействани рискове, както и вземане на необходимите решения и мерки за разрешаване на проблемите и противодействие на рисковете.</p>
1.3.	Ненавременно изпълнение на всяко от задълженията от страна на Сمارт Системс	Забава в цялостното изпълнението на поръчката. Нарушаване на крайните срокове на отделните етапи както и на крайния срок на проекта.	За противодействие на този риск сме предложили план график за изпълнение на дейностите по проекта, чието стриктно спазване ще противодейства успешно на този риск. Този график ще се

			контролира от Ръководителя на проекта и при индикации за потенциалното му нарушаване ще се взимат веднага адекватни мерки с участието на целия екип, участващ в изпълнението на проекта, както от страна на Сمارт Системс , така и от страна на Възложителя
1.4.	Неправилно и неефективно разпределяне на ресурсите и отговорностите при изпълнението на договора	Това е валидно както страна на Сمارт Системс , така и от страна на Възложителя. Възможно е забавяне в спазването на предварително заложените срокове по отделните етапи или в най-лошия случай на крайния срок на проекта.	За избягване на този риск Сمارт Системс и Възложителя трябва непрекъснато да следят нивото на ресурсна обезпеченост на изпълнението на проекта и при нужда да включва допълнителни ресурси. Това ще е контролира от Ръководителя на проекта и при индикации за потенциалното му нарушаване ще се взимат веднага адекватни мерки с участието на целия екип, участващ в изпълнението на проекта, както от страна на Смарт Системс , така и от страна на Възложителя
1.5.	Забавяне при изпълнение на проектните дейности, опасност от неспазване на срока за изпълнение на настоящата поръчка	Опасността от неспазване на срока за изпълнение на настоящата поръчка.	Този риск е реален , предвид на съотношението на обема работа към предлаганото време за изпълнение на поръчката. Най-силните инструменти за противодействие на този риск са: <ul style="list-style-type: none"> • задълбоченият опит на експертите на Смарт Системс , натрупан при разработката и внедряването на уеб базирани информационни

			<p>системи;</p> <ul style="list-style-type: none"> • прилагането на предложената от нас ефективна Организация за изпълнение на проекта. • висока мобилизация на ресурсите от страна както на Смарт Системс , така и на Възложителя.
1.6.	Грешки при разработване на функционалностите на системата	Неправилно функциониране на системата. Проблеми при тестването. Вероятност от неприемане на системата от Възложителя и респективно цялостното неспазване на крайния срок на проекта	<p>Внимателен и подробен анализ на изискванията, който намира отражение в детайлната техническа спецификация (Системен проект), внимателно анализиран и приет от страна на възложителя. За противодействие на този риск, е дефинирано и ще се прилагат ефективни механизми за управление и контрол на качеството. Тясно придържане към спецификацията на системата и ранна валидация от страна на Възложителя също ще допринесат за минимизиране на този риск.</p>
1.7.	Недостатъчна яснота по правната рамка и/или променяща се правна рамка по време на изпълнение на проекта;	Създадената функционалност да интерпретира неправилно нормативната уредба или да бъде несъвместима с промени в правната рамка, които предстоят	<p>Провеждане на работни сесии, анализ на съществуваща документация във връзка с нормативната уредба, навременно оценяваен на въздействието върху проекта при евентуални промени на съществуващата правна рамка</p>
1.8.	Липса на задълбоченост при изследването и описанието на бизнес процесите и данните	Неточно отразяване на изискванията на Възложителя и респективно	<p>Инструментът за противодействие на този риск е задълбоченият анализ и опит на</p>

		наслагване на проблеми в последващата фаза на разработка на системата, което накрая ще доведе до липса на определени функционалности в крайния продукт.	експертите на Сمارт Системс, натрупан при разработката и внедряването на информационни системи. Непрекъсната комуникация с екипа на възложителя от гледна точка на регулярно съгласуване на описанието на процесите и данните.
1.9.	Неинформиране на Възложителя за всички потенциални проблеми, които биха могли да възникнат в хода на изпълнение на дейностите	Невъзможност Възложителя да вземе адекватни предварителни мерки за решаването на тези проблеми, като крайния резултат би могъл да бъде невъзможност за завършване на даден етап и респективно забава в сроковете на изпълнение на проекта.	Противодействието на този риск ще бъде обезпечено чрез анализиране на дейностите по реализация на проекта, идентифициране на потенциалните проблеми и навременното им обсъждане с Възложителя, както и регулярно допълване на списъка с идентифицирани рискове и от двете страни и изготвяне на стратегия за реакция на новоидентифицираните рискове и тяхната превенция. Регулярни междинни отчети за извършената работа.
1.10.	Риск за администриране на системата след изтичане на периода на гаранционна поддръжка	Неработоспособност на системата. Нарушаване на регулярната нормална работа на Възложителя	За противодействие на този риск, през етап „Обучение“ се провеждат обучения на 2 ма системните администратори. Крайният резултат от този етап ще включва обучени системни администратори за администриране на системата, което е критична предпоставка за безпроблемното администриране след изтичане на гаранционната поддръжка.

✓ **Динамично измерение на дейността „Идентифициране на рисковете“**

Тази дейност е повтарящ се (итеративен) похват.

Рискове от Техническата спецификация, допълнително идентифицираните от Сمارт Системс в тръжната фаза и евентуално идентифицираните при подписване на договора за изпълнение рискове ще формират **първата версия Регистъра на рисковете**.

Итеративното изпълнение на тази дейност през целия жизнен цикъл на проекта може да породи **актуализация на Регистъра на рисковете** в края на всеки един от етапите на проекта, определени в т.6. от Техническото задание.

4.7.4.5. Анализ и Планиране на реакцията

✓ **Статично измерение на дейностите „Анализ“ и „Планиране на реакцията“**

Както беше изложено в представянето на Методологията за управление на риска в проекта, **качествения анализ** дава оценка за вероятността за възникване и влиянието на рисковете, а **количествения анализ** дава оценка на ефекта от тези рискови събития. Това е процес на **приоритизация** на вече идентифицираните рискове от предходната дейност, за целите на бъдещ анализ и действия за противодействие.

Планирането на реакцията е дейност, неразривно свързана с анализа и обичайно се извършва едновременно с него. Крайната цел е за всеки идентифициран и приоритизиран риск да се планира реакция, която да е адекватна на важността на риска, ценово ефективна и реалистична в контекста на проекта.

Прилагането на предлаганата Методология за управление на риска по отношение на дейностите „Анализ“ и „Планиране на реакцията“ ще включва итеративното използване на техниките за качествен и количествен анализ и планиране на реакцията с цел определяне на номинални стойности на следните показатели за всеки от идентифицираните рискове, които се регистрират в **таблица „Приоритизация на рисковете и планирани реакции“**, неразделна част **Регистъра на рисковете**:

- Степен на значимост (Влияние)
- Вероятност от настъпване
- Рисковата експозиция
- Индикатор
- Планирани реакции

На база извършените от нас предварителен Анализ (качествен и количествен) и Планиране на реакцията достигнахме до следната първоначалната работна версия на таблицата „Приоритизация на рисковете и планирани реакции“:

Таблица 9 Приоритизация на рисковете и мерки за противодействие

№ на риск	Степен на значимост	Вероятност от настъпване	Рискова експозиция	Мерки за противодействие
-----------	---------------------	--------------------------	--------------------	--------------------------

	(Влияние)			
1.1	0,7	0,10	0,07	Своевременно осигуряване на информация за предстоящи промени в правната рамка, за да може резултатите да бъдат съобразени с тях. Гъвкавост на разработките с възможност за параметризация и вариативност.
1.2	0,8	0,10	0,08	<p>Мерки за противодействие както следва:</p> <ul style="list-style-type: none"> Ръководителят на екипа, подпомаган от Тестовия инженер. Да създадат организацията за изпълнение на всеки етап от проекта, която задължително включва възлагане, следене, координация и съгласуване на изпълнението на дейностите през етапа и осъществяване на непрекъсната комуникация с Възложителя; Регулярност на срещите на управленското ниво на проекта от страна на Сمارт Системс и Възложителя, която включва: Седмични срещи за бърз преглед на прогреса по изпълнение на проекта и синхронизация на дейностите между всички членове на екипа за изпълнение на проекта; <p>Месечни срещи за детайлен преглед на прогреса по изпълнение на проекта, обсъждане на ресурсното обезпечаване на проекта, анализ на възникналите проблеми и задействани рискове, както и вземане на необходимите решения и мерки за разрешаване на проблемите и противодействие на рисковете.</p>
1.3	0,9	0,10	0,09	За противодействие на този риск сме предложили план график за изпълнение на дейностите по проекта, чието стриктно спазване ще противодейства успешно на този риск.
1.4.	0,6	0,10	0,06	За избягване на този риск Смарт Системс и Възложителя трябва непрекъснато да следят нивото на ресурсна обезпеченост на изпълнението на проекта и при нужда да включва допълнителни ресурси. Това ще се контролира от Ръководителя на проекта и при индикации за потенциалното му нарушаване ще се вземат веднага адекватни мерки с участието на целия екип, участващ в изпълнението на проекта, както от страна на Смарт Системс , така и от страна на Възложителя.

1.5.	0,9	0,10	0,09	<p>Този риск е реален, предвид на съотношението на обема работа към предлаганото време за изпълнение на поръчката.</p> <p>Най- силните инструменти за противодействие на този риск са:</p> <ul style="list-style-type: none"> • задълбоченият опит на експертите на „Смарт системс 2010“ ЕООД, натрупан при разработката и внедряването на уеб базирани информационни системи; • прилагането на предложената от нас ефективна Организация за изпълнение на проекта. • висока мобилизация на ресурсите от страна както на Смарт Системс , така и на Възложителя.
1.6.	0,6	0,10	0,06	<p>Внимателен и подробен анализ на изискванията, който намира отражение в детайлната техническа спецификация (Системен проект), внимателно анализиран и приет от страна на възложителя.</p> <p>За противодействие на този риск, е дефинирано и ще се прилагат ефективни механизми за управление и контрол на качеството.</p> <p>Тясно придържане към спецификацията на системата и ранна валидация от страна на Възложителя също ще допринесат за минимизиране на този риск.</p>
1.7.	0,6	0,2	а0,12	<p>Провеждане на работни сесии, анализ на съществуваща документация във връзка с нормативната уредба, навременно оценяваен на въздействието върху проекта при евентуални промени на съществуващата правна рамка</p>
1.8.	0,7	0,10	0,07	<p>Инструментът за противодействие на този риск е задълбоченият анализ и опит на експертите на Смарт Системс , натрупан при разработката и внедряването на информационни системи.</p> <p>Непрекъсната комуникация с екипа на възложителя от гледна точка на регулярно съгласуване на описанието на процесите и данните.</p>
1.9.	0,6	0,10	0,06	<p>Противодействието на този риск ще бъде обезпечено чрез:</p> <p>Анализиране на дейностите по реализация на проекта, идентифициране на потенциалните проблеми и навременното им обсъждане с Възложителя, както и регулярно допълване на списъка с идентифицирани рискове и от двете</p>

				страни и изготвяне на стратегия за реакция на новоидентифицираните рискове и тяхната превенция. Регулярни междинни отчети за извършената работа.
1.10.	0,6	0,10	0,06	За противодействие на този риск, през етап „Обучение“ се провеждат обучения на 2 ма системните администратори. Крайният резултат от този етап ще включва обучени системни администратори за администриране на системата, което е критична предпоставка за безпроблемното администриране след изтичане на гаранционната поддръжка.

По отношение на Рисковата експозиция е определена следната скала за степен на критичност за целите на реакцията:

Таблица 10 - Скала за степен на критичност

Рискова експозиция	Степен на критичност
< 0,1	много ниска
0,1 - 0,3	Ниска
0,3 – 0,5	умерена
0,5 – 0,8	висока
0,8 >	много висока

✓ **Динамично измерение на дейностите „Анализ“ и „Планиране на реакцията“**

Тези дейности са също повтарящ се (итеративен) похват, както и целия процес по управление на риска в проекта.

На база анализа и планирането на реакцията извършени от нас в тръжната фаза и на изпълнението на тези дейности през етапа на проекта „Проектиране“ ще се формира първата версия на таблицата „**Приоритизация на рисковете и планирани реакции**“ като неразделна част от Регистъра на рисковете.

Итеративното изпълнение на тези дейности през целия жизнен цикъл на проекта може да породи **актуализация на таблицата „Приоритизация на рисковете и планирани реакции“** в края на всеки от етапите на проекта, определени в т.6. от Техническата спецификация.

4.7.4.6. Наблюдение и контрол

✓ **Статично измерение на дейността „Наблюдение и контрол“**

Планираните реакции на рисковете, включени в **Регистъра на рисковете** се изпълняват през целия жизнен цикъл на проекта. Ефектът от прилагането на реакциите обаче може да се промени ако не се отчете динамиката на проекта и настъпилите нови обстоятелства и промени в средата като цяло. Това именно обосновава нуждата от дейността Наблюдение и контрол, която при необходимост запалва итеративното прилагане на фазите Анализ и Планиране на реакцията. Това гарантира, че във всеки един момент управлението на риска в проекта се базира на коректна приоритизация на рисковете и адекватни реакции за тези рискове.

✓ **Динамично измерение на дейността „Наблюдение и контрол“**

Тази дейност има най-силно изразения итеративен характер и се изпълнява през целия жизнен цикъл на проекта. Наблюдаване за нови, променящи се или вече остарели рискове ще поражда актуализация на Регистъра на рисковете и итеративно прилагане на останалите дейности по управление на риска – Анализ (качествен и количествен) и Планиране на реакцията.

5. ЕТАПИ НА РЕАЛИЗАЦИЯ НА ДЕЙНОСТИТЕ ПО ПРОЕКТА

5.1. Анализ на данните и изискванията

По време на Етапа „Анализ на данните и изискванията“ Сمارт Системс ще следва Методологията за усъвършенстване на работните процеси за предоставяне на административни услуги и Наръчника за прилагане на методологията, приета с Решение No 578 на Министерския съвет от 30 септември 2013 г.

В процеса на бизнес анализ ще бъде изследвана съвместимостта на бизнес процесите на Възложителя с вече одобрени оптимизирани референтни модели за предоставяне на услуги и нормативни изисквания на Базисен модел за Комплексно административно обслужване в държавната администрация. При наличие на разработени модели за предоставяне на услуги по „Епизоди от живота“ и „Събития от бизнеса“, които включват услуги, предоставяни от Възложителя, ще бъдат съобразени нуждите от модификации в референтните модели, за да се постигне подобряване на времето и намаляване на административната тежест при комплексно обслужване, спрямо предоставянето на отделните услуги поединично;

- Ще бъде предвидена фаза на проучване, по време на която ще бъдат дефинирани потребителските нужди, ще бъдат проведени предварителни тестове с потребители, и ще бъде изработен план, по който да се адресират идентифицираните нужди;
- Ще бъдат предвидени периодични продуктови тествания по време на разработката и внедряването на Системата, с извадка (фокус-група) от бъдещите потребители, чрез които да се изпита и оцени използваемостта на услугите и потребителските интерфейси, както и за да бъдат отстранени затруднения и несъответствия със заданието;
- Ще бъдат спазени нормативните изисквания за еднократно събиране и повторна употреба на данни в държавната администрация (съгласно АПК и ЗЕУ) и в разработените бизнес процеси няма да бъдат изисквани данни, които могат да се извлекат автоматично. При необходимост Сمارт Системс ще предложи на Възложителя адекватни промени в нормативната уредба, които да хармонизират съответните секторни нормативни изисквания с общите разпоредби на Административно процесуалния кодекс, Закона за електронно управление, Закона за електронния документ и електронния подпис и приложимите подзаконови актове, ако действащата нормативна уредба изисква:
 - изрично обявяване на обстоятелства или данни, които се администратират и/или удостоверяват от други държавни органи и могат да бъдат получени по служебен път, включително и автоматизирано през съответни интеграционни интерфейси;

- други нормативни изисквания, които водят до неоптимални или ненужно бюрократични процеси, които биха могли да бъдат оптимизирани при заявяване и предоставяне на електронни административни услуги;
- Потребителският път от влизане на сайта до заявяване и получаване на услуга и пътят от регистрация на нов потребител до заявяване и получаване на услуга ще бъде оптимизиран;
- При оптимизацията на потребителския път ще бъде отчетено всяко действие от страна на потребителя (натискане на бутон, въвеждане на данни, прочитане на текст и пр.), което може да се спести.

5.2. Изготвяне на системен проект

Смарт Системс т ще изготви проект на техническа спецификация на системата, която подлежи на одобрение от Възложителя. В проекта на техническа спецификация ще бъдат описани всички функционални и нефункционални изисквания, включени в проекта. Изготвянето на техническата спецификация включва следните основни задачи:

- Определяне на концепция на информационната система на базата на техническата спецификация;
- Дизайн на информационната система, хардуерната и комуникационната инфраструктура;
- Дефиниране на детайлни изисквания, които трябва да се реализират в Системата;
- Изготвяне на план за техническа реализация;
- Определяне на потребителския интерфейс.

При изпълнението на задачите ще се дефинират модели на бизнес процеси, модели на стандартни справки и анализи, модели на печатни бланки, политика за сигурност и защита на данните, основни изграждащи блокове, транзакции, технология на взаимодействие, мониторинг на системата, спецификация на номенклатурите, роли в системата и други. При документирането на изискванията, с цел постигане на яснота и стандартизация на документите, ще се използва стандартен език за описание на бизнес процеси – BPMN.

Системният проект подлежи на одобрение от Възложителя. В случай на забележки, корекции или допълнения от страна на Възложителя Смарт Системс т ще ги отрази в системния проект в срок не по-късно от 10 работни дни.

Подробно описание на подхода за изпълнение на дейностите по проектиране може да бъде намерено в т. 7.1. „Предложение за извършване на дейностите по анализ и проектиране“

5.3. Разработване на софтуерното решение

Етапът на разработка включва изпълнението на следните задачи:

- Разработка на модулите на информационната система, съгласно изискванията на настоящото техническо задание и системния проект (техническата спецификация);
- Провеждане на вътрешни тестове на Системата (в среда на разработчика);
- Изготвяне на детайлни сценарии за провеждане на приемателните тестове за етапи „Тестване“ и „Внедряване“ на проекта.

Подходът (Методологията) за разработка на софтуерното приложение е описан в т.7.2 „Методики за извършване на дейностите по разработка на системата“. Там са описани и инструментите за разработка и средата за провеждане на вътрешните тестове. Описания подход е адаптиран за успешната реализация на Системата

5.4. Тестване

По време на софтуерната разработка и след нейното приключване Смарт Системс ще извърши обстойно тестване на Системата.

Смарт Системс т ще предвиди време за отстраняване на възникнали проблеми при тестването или несъответствия в разработения софтуер.

Смарт Системс т ще проведе тестване на софтуерното решение в създадената тестова среда, с цел да се удостовери, че разработените програмни продукти са работоспособни и отговарят на изискванията на Възложителя.

В рамките на тази фаза ще бъдат отстранени разминаванията между изискванията и функционалността на системата, както и на откритите програмни грешки. Посредством тестовите (функционални, интеграционни и тестове за производителност) ще се удостоверява изпълнението на изискванията към системата, спрямо Техническото задание.

Подробно описание на методологията за тестване и предварителен план за тестване може да бъде намерен в точка 7.2.3. „Методология за тестване“.

5.5. Внедряване

Смарт Системс ще внедри системата в реална експлоатация в работната среда, осигурена от НСИ. След успешната инсталация на системата в експлоатационната среда на Възложителя ще бъдат проведени тестове преди въвеждането ѝ в експлоатация, целите, на които са да осигурят работоспособността на системата..

Подробно описание на методиката за внедряване се намира в точка 7.2.2. от настоящия документ.

5.6. Обучение

Смарт Системс ще организира и да проведе обучения за всички видове потребители и ползватели, съответно:

- в Националния статистически институт

За провеждане на обученията, Сمارт Системс ще осигури за своя сметка:

- Необходимия хардуер;
- Необходимия софтуер;
- Зала/Зали за провеждане на обученията;
- Учебни материали;
- Лектори.

Смарт Системс ще изготви и съгласува с Възложителя план-програма. Преди провеждане на обученията, Смарт Системс ще изготви и предаде на Възложителя следните документи, които ще бъдат използвани и като обучителни материали:

- Ръководство на администратора;
- Ръководство на потребителя, илюстрирано с графики/скрийншотове на системата;
- Описание на базата данни;
- Описание на софтуерните модули.

Подробно описание на методиката за обучение може да бъде намерено в т. 7.2.4 от настоящия документ.

5.7. Гаранционна поддръжка

Смарт Системс ще извърши за своя сметка гаранционна поддръжка с 36-месечен гаранционен срок, след приемане в експлоатация на системата, за всички нови функционалности и промени в обхвата на настоящия проект.

При необходимост, по време на гаранционния период ще бъдат осъществявани дейности по осигуряване на експлоатационната годност на софтуера и ефективното му използване от Възложителя, в случай че настъпят явни отклонения от нормалните експлоатационни характеристики, заложиени в системния проект.

Смарт Системс ще предоставя услугите по гаранционна поддръжка, като предоставя за своя сметка единна точка за достъп за приемане на телефонни и e-mail съобщения.

Приоритетите на проблемите се определят от Възложителя в зависимост от влиянието им върху работата на администрацията. Редът на отстраняване на проблемите се определя в зависимост от техния приоритет.

Минималният обхват на поддръжката ще включва:

- Извършване на диагностика на докладван проблем с цел осигуряване на правилното функциониране на системите и модулите;
- Отстраняване на дефектите, открити в софтуерните модули, които са модифицирани или разработени в обхвата на проекта;
- Консултации за разрешаване на проблеми по предложената от Смарт Системс конфигурация на средата (операционна система, база данни, middleware, хардуер и мрежи), използвана от приложението, включително

промени в конфигурацията на софтуерната инфраструктура на мястото на инсталация;

- Възстановяването на системата и данните при евентуален срив на системата, както и коригирането им в следствие на грешки в системата;
- Експертни консултации по телефон и електронна поща за системните администратори на Възложителя за идентифициране на дефекти или грешки в софтуера;
- Актуализация и предаване на нова версия на документацията на системата при установени явни несъответствия с фактически реализираните функционалности, както и в случаите, в които са извършени действия по отстраняване на дефекти и грешки, в рамките на гаранционната поддръжка.

6. СЪОТВЕТСТВИЕ С ОБЩИТЕ ИЗИСКВАНИЯ ЗА ИНФОРМАЦИОННИ СИСТЕМИ В ДЪРЖАВНАТА АДМИНИСТРАЦИЯ

6.1. Функционални изисквания

6.1.1. Интеграция с външни информационни системи

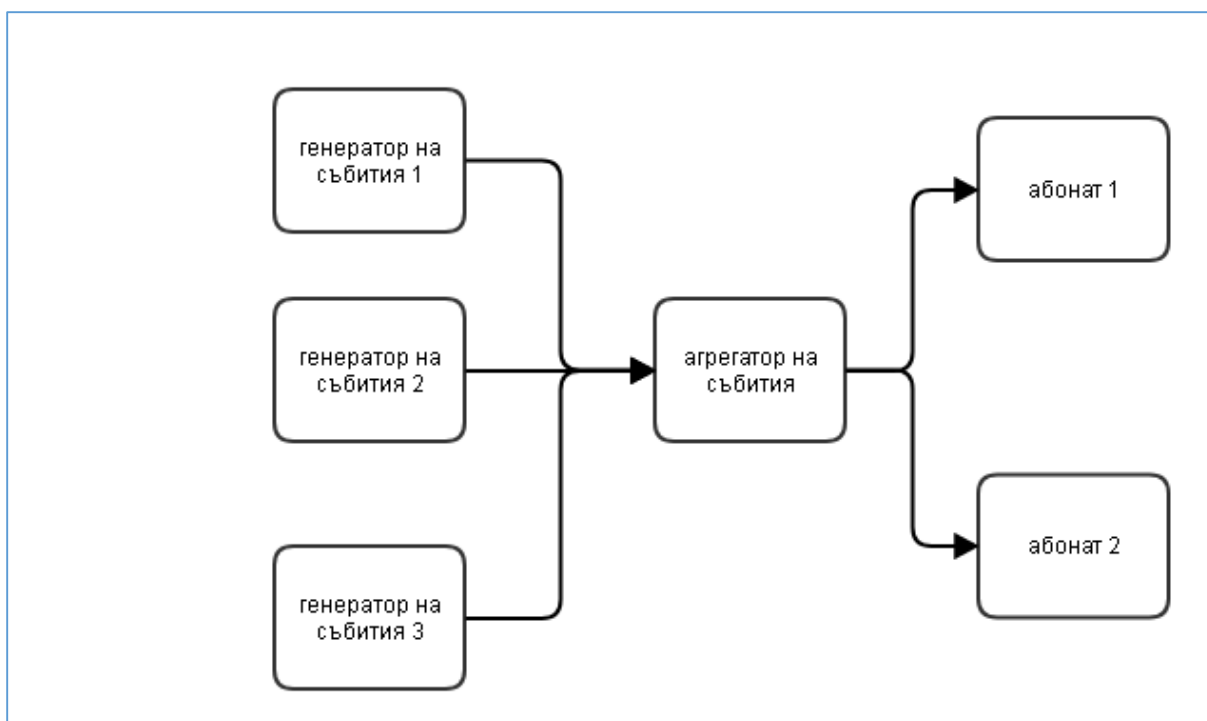
Смарт Системс ще осигури връзка на ИСВТ с вътрешните в НСИ системи – ИСРСЕ, ИСБС, СОПТ, БД „МДЕ”. Актуалните данни за статистическите единици, ще бъдат зареждани от СОПТ и РСЕ. Зареждането може да се изпълнява многократно за указан период от време.

6.1.2. Интеграционен слой

Интеграциите с вътрешните информационни системи в НСИ ще бъде реализирана чрез стандартен интеграционен слой, чрез зареждането им в междинна зона или вътрешна услуга (service) – конкретната реализация ще бъде предоставена за одобрение от НСИ. Интеграциите във вид на услуги между контактните точки ще бъдат съобразени и ще отговарят на действащите изисквания за оперативна съвместимост.

В процеса на обработка на данни в системата се стига до изпълняване на условия, при които следва да се извърши трансфер на данни към външни системи. В последващото описание разглеждаме тази съвкупност от условия като "интеграционни събития", а обменът на данни с външна система – като "интеграционен трансфер"

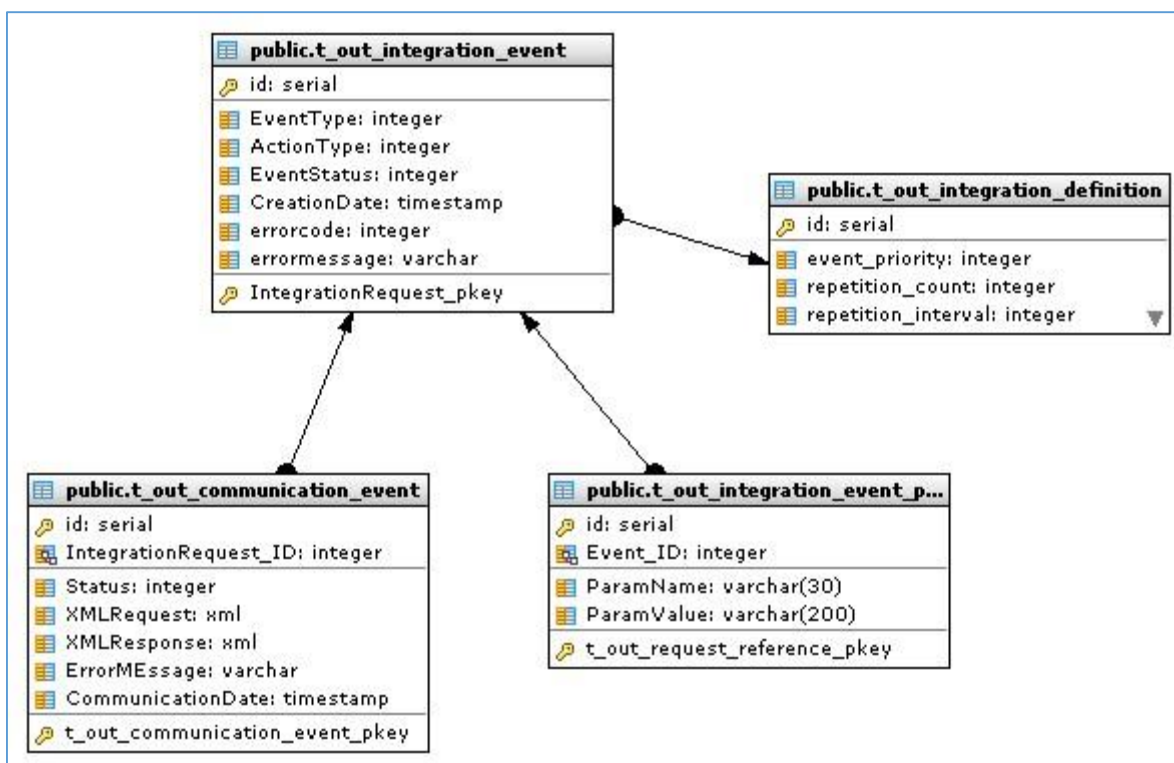
Архитектурата на модулет за интеграция с външни системи е базирана на модела "Агрегатор на събития". Моделът "Агрегатор на събития" се опитва да преодолее ограничаването на традиционния подход за управление на събития, като предостави централно място за публикуване и абониране за събития, които не са нищо друго освен събитие за събития. Агрегаторът за събития се грижи за регистрирането, отписването и извикването на събития, които свободно свързват генераторите на събития и абонатите. На диаграмата по-долу е показана принципна схема на типичен агрегатор на събития в приложение или система.



Фигура 13 Принципна схема на агрегатор на събития в система

Услугата EventAggregator е преди всичко контейнер за събития, които позволяват отделяне на връзките между генераторите и абонатите, така че те да могат да се развиват самостоятелно. Това отделяне е полезно при модулни приложения, защото могат да се добавят нови модули, които отговарят на събитията, дефинирани от ядрото или от други модули.

В контекста на интеграцията на системата с външни системи моделът за агрегиране на събитията ще се реализира по следния начин: генерирането на събития ще е достъпно в Системата посредством специализирано API, даващо възможност за създаване и съхраняване на събития в момента на възникването им, като моделът на данните позволява да се въведат неограничен брой параметри-референции, отразяващи контекста на събитието.



Фигура 14 Модел на данни на общия интеграционен компонент.

На ниво СУБД генерираните събития се съхраняват в таблица **t_out_integration_event**, като в таблица **t_out_integration_event_parameters** се съхраняват параметри-референции във формат ключ/стойност. Параметрите позволяват на модулта за интеграция да възстанови контекста на данните при възникването на събитието– в най-общия случай като параметри се използват уникални идентификатори на информационните обекти в базата данни.

Отделните специфични интеграции към всяка външна система се реализират като програмен код (клас), имплементиращ стандартен интерфейс по отношение на агрегатора на събития, и съдържащ специфична част за осъществяване на конкретната интеграция. Специфичната част най-често представлява web/wcf клиент или REST API комуникатор, като конкретната имплементация е независима и архитектурата позволява добавянето на бъдещи интеграции.

Диспечер на задачите обработва постъпилите събития, като в зависимост от техния тип и допълнителните метаданни от таблица **t_out_integration_definition** приоритизира събитията, определя типа абонат (интегратор) и извикват стандартния дефиниран в интерфейса метод за извършване на интеграцията – генериране на дейта трансфер обекти, извикване на външни уеб услуги със съответната автентификация и отразяване на резултата от комуникацията с външната система.

Самият процес на комуникация, генерираните XML или JSON пакети и възникналите грешки се отразяват в таблица **t_out_communication_event**.

Диспечерът на задачите по интеграцията е имплементиран под формата на системна услуга на операционната система.

Към модульът за интеграция ще се реализира и набор от споделен между отделните интеграции програмен код с общо предназначение – библиотеки, помощни класове и като цяло инфраструктура за работа с уеб услуги, автентикация със и управление на сертификати, криптиране и тунелиране.

Модульът за интеграция ще предоставя потребителски интерфейс на системните администратори за преглед на текущия статус на комуникациите, изпълнените събития и възникналите грешки, както и справки за изминали периоди.

Основните предимства на избрания подход са следните:

- реализацията на интеграцията е отделена от бизнес логиката за съхраняване и управление на основните данни на Системата
- процесът на обмен на данни с външните системи може да се извършва в асинхронен режим. В случай на загуба на свързаност с външните системи интеграционните събития се запазват и трансферът на данните се осъществява при възстановяване на свързаността.
- налице е пълна хронологична проследяемост на трансфера на данните и възникналите грешки
- при евентуално бъдещо добавяне на нови външни системи голямата част от програмният код остава непроменена и изтествана (API за генериране на събития и диспечера за задачите) което намалява значително разходите за подобни дейности.

6.1.3. Технически изисквания към интерфейсите

Приложните програмни интерфейси ще отговарят на следните архитектурни, функционални и технологични изисквания:

- Служебните онлайн интерфейси ще бъдат предоставени като уеб-услуги (web-services) и ще осигуряват достатъчна мащабируемост и производителност за обслужване на синхронни заявки (sync pull) в реално време, с максимално време за отговор на заявки под 1 секунда за 95% от заявките, които не включват запитвания до регистри и външни системи.
- Смарт Системс ще обоснове прогнозирано натоварване на Системата и ще предложи критерии за оценка на максимално допустимото време за отговор на машинна заявка. Критерият за оценка ще се основава на анализ на прогнозираното натоварване и на наличния хардуер, който ще се използва. Във фазата на анализ и проектиране, след анализ на обемите данни и планираната физическа архитектура Смарт Системс ще представи обосновано предложение за минималното време за отговор на заявка на базата на посочените по-горе критерии и да осигури нужните условия за спазването му.
- Всички публични и служебни онлайн интерфейси ще поддържат режим “push” или “pull”, в асинхронен и синхронен вариант. На етап бизнес-анализ ще бъде определено

практическото прилагане на всяка от комбинациите, съобразена с реални казуси (use cases), които всеки интерфейс обслужва.

- Ще бъде реализирано интегриране на модул за разпределен кохерентен кеш (Distributed Caching) на „горещите данни“, които Системата получава и/или които се обменят през служебните онлайн интерфейси, като логиката на Системата ще гарантира кохерентност (Cache Coherency) между кешираните данни и данните, съхранявани в базите данни;

Потребителският интерфейс на системата ще използва опростена навигационна структура, базирана на едно меню за достъп до функционалностите на системата, филтрирано според потребителския профил на потребителя и със структура базирана на MenuProvider клас, позволяващ лесно управление и преконфигуриране на системното меню от административния панел.

6.1.4. Електронна идентификация на потребителите

- Системата ще поддържа и стандартен подход за регистрация на потребители с потребителско име и парола – за вътрешни потребители на системата. ИСВТ няма да се достъпва от външни потребители.
- Процесът по регистрация на потребители ще бъде максимално опростен и бърз, но ще включва следните специфични стъпки:
 - Визуализиране на информация относно стъпките по регистрация и информация във връзка с процеса за потвърждаване на регистрацията и активиране на потребителския профил.
 - Избор на потребителско име с контекстна валидация на полетата (in-line validation), включително и за избраното потребителско име;
 - Избор на парола с контекстна валидация на полето (in-line validation) и визуализиране на сложността на паролата като "слаба", "нормална" и "силна";
 - Потребителски пароли съгласно изискванията на Политиката за управление на паролите на НСИ;

За вътрешните потребители на ИСВТ ще бъдат дефинирани типови потребителски групи (роли), на които ще бъдат предоставени съответните права, а отделните потребители ще се включват/изключват в/от тези групи и ползват правата на групите, към които текущо принадлежат. Различните роли ще съответстват на типовете функционални потребители и ще притежават различни набори от права, в зависимост от отговорностите и правомощията на съответните потребители. Ролите ще бъдат изготвени спрямо предоставената таблица:

Име на ролята	Име на ролята Описание на ролята
---------------	----------------------------------

SFT_COMMON	<p>Тази роля съдържа общи права за всички потребители на модул „Микроданни”. Правата включват права за четене на всички статистически класификации и права за изпълнение на стандартните сървърни процедури за дефиниране на контекста в БД и за проверка на правата на потребителя.</p> <p>*** Тази роля не е предназначена да бъде директно предоставяна на потребителите. Вместо това тази роля е GRANT-ната на другите дефинирани роли и се получава от потребителите автоматично при получаване на някоя от другите роли</p>
ISFT_STATISTICIAN	<p>Тази роля е предназначена за потребители, които извършват анализ, редактиране и обработване на микроданни за външнотърговските трансакции. Ролята включва следните права:</p> <ul style="list-style-type: none"> • Всички права на ролята ISFT_COMMON; • Права за четене, писане и редактиране в Първичните регистри на наблюдения; • Права за въвеждане на предложения за промяна лимитите за единични тегла и цени на стоки от Комбинираната номенклатура; • Права за стартиране на контролни справки върху обработките на първичните микроданни; • Права за автоматично коригиране, обработване, анулиране, де-анулиране и сторниране на отделен запис с първични микроданни; • Права за четене на Крайните регистри на наблюдения и на данните за търговците, съхранявани в ИСВТ; • Права за четене от таблиците с данни за статистически класификации и конфигурационни данни за системата.
ISFT_SPEC_GOODS	<p>Тази роля е предназначена за потребители, които освен анализ, редактиране и обработване на микроданни за външнотърговските трансакции, извършват въвеждане и обработване на данни за движението на специфични стоки. Ролята включва следните права:</p> <ul style="list-style-type: none"> • Всички права на ролята ISFT_STATISTICIAN; • Права за четене, въвеждане, редактиране и обработване на данни от отчети за специфични стоки и движения.

ISFT_ESTIMATION	<p>Тази роля е предназначена за потребители, които изготвят оценки на не отговорили търговци и не обхванати потоци по ИНТРАСТАТ. Ролята включва следните права:</p> <ul style="list-style-type: none"> • Всички права на ролята ISFT_STATISTICIAN; • Права за въвеждане, редактиране и обработване на данни за оценка по ИНТРАСТАТ.
ISFT_TECHNOLOG	<p>Тази роля е предназначена за потребители, които зареждат входни файлове с първични микроданни, извършват обработване и редактиране на проблемите в тях, генерират изходни данни-файлове за външни потребители и други системи в НСИ. Поддържат общосистемната информация, необходима за обработките на първични данни. Контролират обмена на данни с други системи в НСИ. Ролята включва следните права:</p> <ul style="list-style-type: none"> • Всички права на ролята ISFT_STATISTICIAN и ISFT_ESTIMATION; • Права за зареждане на входни файлове с първични микроданни, предоставени от АМ и НАП; • Права за стартиране на групови обработки върху първични микроданни (автоматични корекции и пакетна обработка); • Права за сторниране на месечни данни от АМ и обработване на годишни данни; • Права за генериране от Крайния регистър на изходни данни-файлове; • Права за актуализиране на данните за търговците в ИСВТ с данни от СОПТ и обратно; • Права за зареждане на файлове с данни за статистически класификации и връзки между елементите на различни класификации; • Права за ръчно въвеждане и редактиране на данни за статистически класификации и общосистемната информация, необходима при обработване на първични микроданни; • Права за зареждане на файлове с данни за лимити на тегла и единични цени на стоки от Комбинираната номенклатура; • Права за редактиране, одобряване и отхвърляне на предложения за промяна на лимити на тегла и

	единични цени на стоки от Комбинираната номенклатура.
ISFT_POWER_USER	<p>Тази роля е предназначена за потребители, които трябва да имат най-големи правомощия при поддържане и обработване на данни за външно търговските потоци. Ролята включва следните права:</p> <ul style="list-style-type: none"> Всички права на ролите ISFT_TECHNOLOG и ISFT_SPEC_GOODS.
ISFT_ADMI	<p>Тази роля е предназначена за администраторите на модул „Микроданни”. Ролята включва следните права:</p> <ul style="list-style-type: none"> Всички права на ролята ISFT_TECHNOLOG; Права за регистриране и промяна на данни за организационната структура на НСИ; Права за регистриране и промяна на данни за потребители и роли; Права за предоставяне/отнемане на права на потребители и роли; Права за промяна стойностите на конфигурационните параметри и системните съобщения, съхранявани в таблици в БД.
ISFT_ORT	<p>Тази роля е предназначена за потребители, които поддържат Статистически общ регистър на търговците в НСИ и включва следните права:</p> <ul style="list-style-type: none"> Всички права на ролята ISFT_COMMON; Права за четене на данните за търговците, съхранявани в структурите на ИС „Външна търговия”; Права за актуализиране на данните за търговците в ИСВТ с данни от СОРТ и обратно – обновяване на СОРТ с данни за натрупаните от ИСВТ обороти на търговци.

Предоставянето/отнемането на роли на/от отделните потребители ще се извършва от администраторите на системата през специализиран контролен панел.

6.1.5. Формиране на изгледи

Потребителите на Системата ще получават разрези на информацията чрез добавени възможности за филтриране, пренареждане и агрегиране на данните. Резултатите ще бъдат представяни чрез:

- Визуализиране на таблици;

- Графична визуализация на екран;
- Разпечатване на хартиен носител;
- Експорт на данни в един или в няколко от изброените формати – ODF, Excel, PDF, HTML, TXT, XML, CSV.

6.1.6. Администриране на Системата

Системата ще разполага с функционалност осигуряваща администриране на потребителите и правата за достъп.

Тази функционалност ще позволява на Администраторите на системата да извършват следните дейности по администриране на потребители на системата:

- Регистриране на нов потребител;
- Деактивиране на потребител;
- Активиране на потребител;
- Предоставяне/отнемане на права на достъп на потребител;
- Промяна на парола на потребител (може да се изпълнява и от съответния потребител).
- Дефиниране на потребителски роли

6.2. Нефункционални изисквания към информационната система

6.2.1. Авторски права и изходен код

- Всички компютърни програми, които се разработват за реализиране на системата, ще отговарят на критериите и изискванията за софтуер с отворен код.
- Изключителното право на собственост принадлежи на Възложителя - всички авторски и сродни права върху произведения, обект на закрила на Закона за авторското право и сродните му права, включително, но не само, компютърните програми, техният изходен програмен код, структурата и дизайнът на интерфейсите и базите данни, чието разработване е включено в предмета на поръчката, възникват за Възложителя в пълен обем без ограничения в използването, изменението и разпространението им и представляват произведения, създадени по поръчка на Възложителя съгласно чл. 42, ал. 1 от Закона за авторското право и сродните му права.
- Приложимите и допустими лицензи за софтуер с отворен код са:
 - GPL (General Public License) 3.0
 - LGPL (Lesser General Public License)
 - AGPL (Affero General Public License)
 - Apache License 2.0
 - New BSD license

- MIT License
- Mozilla Public License 2.0

За създаване на софтуерното решение, Сمارт Системс т ще използва базови софтуерни платформи с отворен код, които имат разработена техническа документация за актуалната стабилна версия, възможност за предоставяне на комерсиална поддръжка и са подкрепени от организации с идеална цел или комерсиални организации, а именно Microsoft .NET Framework.

- Изходният код (Source Code), разработван по проекта, както и цялата техническа документация ще бъдат публично достъпни онлайн като софтуер с отворен код от първия ден на разработка чрез използване на система за контрол на версиите и хранилището по чл. 7в, т.18 от ЗЕУ;

Проектът ще използва публично достъпни софтуерни библиотеки с отворен код за визуализация и подобряване на усещанията на потребителя.

Ще се изследва възможността резултатният продукт (системата) да се изгради частично (библиотеки, пакети, модули) или изцяло на базата на съществуващи софтуерни решения, които са софтуер с отворен код. Когато е финансово оправдано, ще се предпочита този подход пред изграждането на собствено софтуерно решение в цялост, от нулата.

- Предвижда се използването на Система за контрол на версиите и цялата информация за главното копие на хранилището, прието за оригинален и централен източник на съдържанието, ще бъде достъпна публично, онлайн, в реално време. По-подробно описание може да бъде намерено в т.7.2.1.4 “ Процедура за управление на програмния код „ от настоящото предложение.

6.2.2. Системна и приложна архитектура

По отношение на системната архитектура приложението ще спазва следните изисквания посочени от Възложителя в техническата спецификация:

- Системата ще бъде реализирана като разпределена модулна информационна система. Системата ще бъде реализирана със стандартни технологии и ще поддържа общоприети комуникационни стандарти, които ще гарантират съвместимост на системата с бъдещи разработки. Съществуващите модули и функционалности ще бъдат рефакторирани и/или надградени по начин, който да осигури изпълнението на настоящето изискване.
- Бизнес процесите и услугите ще бъдат проектирани колкото се може по-независимо с цел по-лесно надграждане, разширяване и обслужване. Системата ще е максимално параметризирана и ще позволява настройка и промяна на параметрите през служебен (администраторски) потребителски интерфейс.
- Ще бъде реализирана функционалност за текущ мониторинг, анализ и контрол на изпълнението на бизнес процесите в системата.
- При разработката, тестването и внедряването на системата, Смарт Системс т ще прилага наложени се архитектурни (SOA, MVC или еквивалентни) модели и дизайн-

шаблони, както и принципите на обектно ориентирания подход за разработка на софтуерни приложения.

- Системата ще бъде реализирана със софтуерна архитектура, ориентирана към услуги – Service Oriented Architecture (SOA).
- Взаимодействията между отделните модули в системата и интеграциите с външни информационни системи ще се реализират и опишат под формата на уеб-услуги (Web Services), които ще са достъпни за ползване от други системи в държавната администрация, а за определени услуги - и за гражданите и бизнеса;
- За всеки от отделните модули/функционалности на системата ще се реализират и опишат приложни програмни интерфейси – Application Programming Interfaces (API). Приложните програмни интерфейси бъдат достъпни и за интеграция на нови модули и други вътрешни или външни системи;
- Приложните програмни интерфейси и информационните обекти задължително ще поддържат атрибут за версия;
- Версията на програмните интерфейси, представени чрез уеб-услуги, ще поддържа версията по един или няколко от следните начини:
 - като част от URL-а;
 - като GET параметър;
 - като HTTP header (Асепт или друг).
- За отделните приложни интерфейси ще бъде разработен софтуерен комплект за интеграция (SDK) на поне две от популярните развойни платформи - .NET, Java, PHP.
- Системата ще осигурява възможности за разширяване, резервиране и балансиране на натоварването между множество инстанции на сървъри с еднаква роля;
- При разработването на системата ще се предвидят възможни промени, продиктувани от непрекъснато променящата се нормативна, бизнес и технологична среда. Системата ще бъде разработена като гъвкава и лесно адаптивна, като отчита законодателни, административни, структурни или организационни промени, водещи до промени в работните процеси.
- Смарт Системс т ще осигури механизми за реализиране на бъдещи промени в системата без промяна на съществуващия програмен код. Когато това не е възможно, времето за промяна, компилиране и пускане в експлоатация ще е сведено до минимум. Бъдещото развитие на Системата ще се налага във връзка с промени в правната рамка, промени в модела на работа на потребителите, отстраняване на констатирани проблеми и др. Такива промени ще се извършват през целия период на експлоатация на регистъра, включително и по време на гаранционния период.
- Архитектурата на системата и всички софтуерни компоненти (системни и приложни)

ще бъдат така подбрани и разработени, че да осигуряват работоспособност и отказоустойчивост на Системата, както и недискриминационно инсталиране (без различни условия за инсталиране върху физическа и виртуална среда) и опериране в продуктивен режим, върху виртуална инфраструктура, съответно върху Държавния хибриден частен облак (ДХЧО).

- Смарт Системс т ще проектира, подготви, инсталира и конфигурира като минимум следните среди за системата: тестова, Sandbox, продуктивна.
- Системата ще бъде разгърната върху съответните среди (тестова за вътрешни нужди, тестова за външни нужди, Sandbox и продуктивна);
- Тестовата среда за външни нужди ще бъде създадена и поддържана като "Sandbox", така че да е достъпна за използване и извършване на интеграционни тестове от разработчици на информационни системи, включително такива, изпълняващи дейности за други администрации или бизнеса, с цел по-лесно и устойчиво интегриране на съществуващи и бъдещи информационни системи. Тестовата среда за външни нужди ще е напълно отделна от останалите среди и нейното използване няма да влияе по никакъв начин на нормалната работа на останалите среди или да създава каквито и да било рискове за информационната сигурност и защитата на личните данни;
- Мрежата на държавната администрация (EECM) ще бъде използвана като основна комуникационна среда и като основен доставчик на защитен Интернет капацитет (Clean Pipe) – изискванията на софтуерните компоненти по отношение на използвани комуникационни протоколи, TCP портове и пр. Трябва да бъдат детайлно документирани от Смарт Системс, за да се осигури максимална защита от хакерски атаки и външни прониквания чрез прилагане на подходящи политики за мрежова и информационна сигурност от Възложителя в инфраструктурата на Държавния хибриден частен облак и EECM;
- НСИ разполага със среда за виртуализация изградена с продукта VMware vCenter Server 6.5. и притежава лицензи за операционни системи Windows Server 2016 R2 x64 Standard, база данни Microsoft SQL Server 2016 x64 Standard или по високи. Смарт Системс ще използва посочените софтуерни продукти и лицензи при реализацията.
- При планиране и реализация на отделните компоненти на системната и приложната архитектура на Системата Смарт Системс ще прилага следните добри практики:
 - ще използва утвърдения MVC дизайн-шаблон;
 - ще спазва препоръките на международните стандарти на W3C относно визуализация и достъпност на информацията в Интернет;
 - ще спазва изискванията за интернационализация, като използва само UTF-8 кодиране на данните, съхранява текстовете от потребителския

интерфейс извън програмния код и има разработени средства за превод и превключване на езиковите версии на интерфейса;

- ще реализира обмен на данни с външни системи (където е приложимо) на базата на архитектура, ориентирана към услуги чрез REST протокол;
 - ще предоставя данни, за които е първоизточник, в отворен, стандартен машинно-четим формат, напр. xml, csv;
 - ще използва системи за пълно текстово търсене;
 - ще разработи процедури за архивиране, възстановяване и възпроизвеждане на системните функции и данните в случай на неизправност, авария или бедствие.
- За реализиране на търсенето ще бъде използвана система за пълно текстово търсене, а именно Solr/Lucene.
 - Ще бъде създаден административен интерфейс, чрез който може да бъде извършвана конфигурацията на софтуера.
 - За всеки обект в Системата се предвижда да има уникален идентификатор.
 - Записите в регистрите няма да подлежат на изтриване или на промяна, а всяко изтриване или промяна ще представлява нов запис.

Детайлно описание на подхода за реализация е налично в 7.2. „Предложение за извършване на дейностите по разработка на системата“ от настоящото предложение.

6.2.3. Повторно използване (преизползване) на ресурси и готови разработки

Проектът ще преизползва максимално налични публично достъпни инструменти, библиотеки и платформи с отворен код, като за реализацията на Системата ще се използват максимално софтуерни библиотеки и продукти с отворен код.

Настоящият проект позволява при реализация на дадена техническа функционалност да бъдат използвани един или няколко от съществуващите множество отворени алтернативни проекти, като с приоритет се ползват тези проекти, които са финансирани със средства на Европейския съюз, както и на такива, в които Сمارт Системс има активни разработчици. В случаите, когато липсва подходяща open source алтернатива с необходимата функционалност, се допуска използване на closed source и на инструменти, библиотеки, продукти и системи с платен лиценз, но това става за сметка на Смарт Системс, който трябва да осигури поддръжка от комерсиална организация, развиваща основните отворени продукти, които ще бъдат използвани като минимум за операционните системи и софтуерните продукти за управление на базите данни.

Отворените проекти ще отговарят на следните критерии:

- За разработката им да се използва система за управление на версиите на кода и да е наличен механизъм за съобщаване на несъответствия и приемане на допълнения;
- Да имат разработена техническа документация за актуалната стабилна версия;
- Да имат повече от един активен програмист, работещ по развитието им;
- Да имат възможност за предоставяне на комерсиална поддръжка;
- Да нямат намаляваща от година на година активност;
- По възможност проектите да са подкрепени от организации с идеална цел, държавни или комерсиални организации;
- По възможност проектите да имат разработени unit tests с code coverage над 50%, а проектът да използва Continuous Integration (CI) подходи – build bots, unit tests run, регулярно използване на статични/динамични анализатори на кода и др.

Смарт Системс не идентифицира свободните компоненти и средства, които възнамерява да използва.

Подход за работа с външните софтуерни ресурси

При използването на свободни имплементации на софтуерни библиотеки ще бъде организирано копие (fork) на съответното хранилище в общото хранилище за проекти с отворен код, финансирани с публични средства в България (към момента <https://github.com/government.bg>). Използващите свободните библиотеки компоненти задават за "upstream repo" хранилищата в областта government.bg, като задължително ще бъде реферирана използваната версия/commit identifier.

Когато се налага промяна в изходния код на използван софтуерен компонент, промените ще се извършват във fork хранилището на government.bg в съответствие с изискванията на основния проект. Смарт Системс т ще извърши необходимите действия за включване на направените промени в основния проект чрез "pull requests" и извършване на необходимите изисквани от разработчиците на основния проект промени до приемането им. Тези дейности ще бъдат извършвани по време на целия проект.

При установяване на наличие на нови версии на използваните проекти ще бъде извършван анализ на влиянието върху настоящата система. В случаите, при които се оптимизира използвана функционалност, отстраняват се пропуски в сигурността, стабилността или бързодействието, новата версия се извлича и използва след успешното изпълнение на интеграционните тестове.

6.2.4. Изграждане и поддръжка на множество среди

Смарт Системс ще изгради и ще поддържа минимум следните логически разделени среди:

Development - Чрез Development средата се осигурява работата по разработката, усъвършенстването и развитието на Системата. В тази среда са налични и допълнителните софтуерни системи и инсталации, необходими за управление на разработката – continuous integration средства, системи за автоматизирано тестване и др.

Staging - Чрез Staging средата се извършват тестове преди разгръщане на нова версия от Development средата върху Production средата. В нея се извършват всички интеграционни тестове, както и тестовите за натоварване.

Sandbox Testing - чрез Sandbox средата всички, които трябва да се интегрират към Системата, могат да тестват интеграцията си, без да застрашават работата на продукционната среда.

Production - Това е средата, която е публично достъпна за реална експлоатация и интеграция със съответните външни системи и услуги.

Управлението на средите ще става чрез автоматизирана система за провизиране и разгръщане на системните компоненти. При необходимост от страна на Възложителя Сمارт Системс т ще съдейства за изграждането на нови системни среди.

6.2.5. Процес на разработка, тестване и разгръщане

Процесите, свързани с развитието на Системата, ще гарантират висока прозрачност и възможност за обществен контрол над всички разработки по проекта. Изграждането на доверие в гражданите и в бизнеса налага радикално по-висока публичност и прозрачност чрез отворена разработка и публикуването на системите компоненти под отворен лиценз от самото начало на разработката.

Всички софтуерни приложения, системи, подсистеми, библиотеки и компоненти, които са необходими за реализацията на Системата, ще бъдат разработвани като софтуер с отворен код и ще бъдат достъпни в публично хранилище, поддържани от ДАЕУ съгласно чл. 7в, т. 18 от ЗЕУ и чл. 58 от Наредбата за общите изисквания към информационните системи, регистрите и електронните административни услуги. Към настоящия момент ще се използва общото хранилище за проекти с отворен код, финансирани с публични средства в България (към момента <https://github.com/governmentbg>).

В случай че върху част от компонентите, нужни за компилация, има авторски права, те ще бъдат или в отделно хранилище с подходящия за това лиценз или за тях ще бъде предоставен заместващ „mock up“ компонент, така че да не се нарушава компилацията на проекта.

За всеки един разработван компонент Сمارт Системс ще покрие следните изисквания за гарантиране на качеството на извършваната разработка и на крайния продукт:

- Документиране на Системата в изходния код, минимум на ниво процедура/функция/клас;
- Покритие на минимум 50% от изходния код с функционални тестове
- Използване на continuous integration практики;
- Използване на dependency management.

Във всеки един компонент на Системата, който се build-ва и подготвя за инсталация (deployment), се предвижда да присъстват следните реквизити:

- Дата и час на build;

- Място/среда на build;
- Потребител извършил/стартирал build процеса;
- Идентификатор на ревизията от кодовото хранилище на компонента, срещу която се извършва build-ът.

Ще се анализират възможностите за включване на граждани в процесите по разработка, тестване и идентифициране на пропуски на софтуера. Във фазата на анализ участникът ще предложи механизъм и процедури за реализирането на такива процеси.

6.2.6. Бързодействие и мащабируемост

6.2.6.1. Контрол на натоварването и защита от DoS/DDoS атаки

- На приложно ниво Системата ще поддържа "Rate Limiting" и/или "Throttling" на заявки от един и същ клиентски адрес, както към страниците с уеб-съдържание, така и по отношение на заявките към приложните програмни интерфейси, които са достъпни публично или служебно като уеб-услуги (Web Services) и служебни интерфейси.
- Лимитите за отделни страници, уеб-услуги и ресурси, които се достъпват с отделен URL/URI ще могат да се конфигурират от администраторите на Системата.

6.2.6.2. Бързодействие

При визуализация на уеб-страници системите ще осигуряват висока производителност и минимално време за отговор на заявки - средното време за заявка ще бъде по-малко от 1 секунда, с максимум 1 секунда стандартно отклонение за 95% от заявките, без да се включва мрежовото време закъснение (Network Latency) при транспорт на пакети между клиента и сървъра.

ИСВТ ще позволява едновременно работа в реално време за всички потребители, без това да оказва влияние на работоспособността и скоростта на обработка на данните.

За да се гарантира бързодействието на системата се предвижда създаването на тестове за натоварване.

6.2.6.3. Използване на HTTP/2

С оглед намаляване на служебния трафик, времената за отговор и натоварването на сървърите ще се използва HTTP/2 протокол при предоставяне на публични потребителски интерфейси с включени като минимум следните възможности:

- Включена header compression;
- Използване на brotli алгоритъм за компресия;
- Включен HTTP pipelining;

- HTTP/2 Server push, приоритизиращ специфични компоненти, изграждащи страниците (CSS, JavaScript файлове и др.);
- Публичните потребителски интерфейси трябва да поддържат адаптивен избор на TLS cipher suites според вида на процесорната архитектура на клиентското устройство - AES-GCM за x86 работни станции и преносими компютри (с налични AES-NI CPU разширения), и ChaCha20/Poly1305 за мобилни устройства (основно базирани на ARM процесори);
- Ако клиентският браузър/клиент не поддържа HTTP/2, ще бъде предвиден fallback механизъм към HTTP/1.1. Тази възможност ще може лесно да се реконфигурира в бъдеще и да отпадне, когато браузърите/клиентите, не поддържащи HTTP/2, станат незначителен процент.

HTTP/2 е новата версия на основния протокол за пренос на данни в Уеб - HTTP. HTTP/2 е с повишена производителност, понижено потребление на трафик и нова, оптимизирана организация на комуникацията клиент-сървър. С оглед намаляване на служебния трафик, времената за отговор и натоварването на сървърите, Сمارт Системс т ще използва HTTP/2 протокол при предоставяне на публични потребителски интерфейси.

Едно от основните подобрения в HTTP/2 е мултиплексването на комуникацията клиент-сървър. Предаваните данни се подреждат в пакети (фреймове), които се придвижват в двете посоки в паралелни потоци (стриймове). Това от своя страна позволява на комуникацията да се проведе в една единствена TCP връзка, която може да има множество потоци, с множество пакети. За разлика от HTTP/1.1 където, за да се постигне по-добра производителност, уеб браузърът отваря по няколко TCP връзки със сървъра.

Другите значителни подобрения са компресиране на хедърите, бинарното естество на протокола и самоинициативното подаване на данни от уеб сървъра. В резултат на тези подобрения, зареждането на уеб сайтовете през HTTP/2 е ускорено в пъти, което го прави предпочитан в реализираните от Смарт Системс проекти.

Предвижда се включването минимум на следните възможности:

- Включена header compression;

Компресирането на header-а подобрява ефективността на мрежовото предаване, качеството и скоростта с:

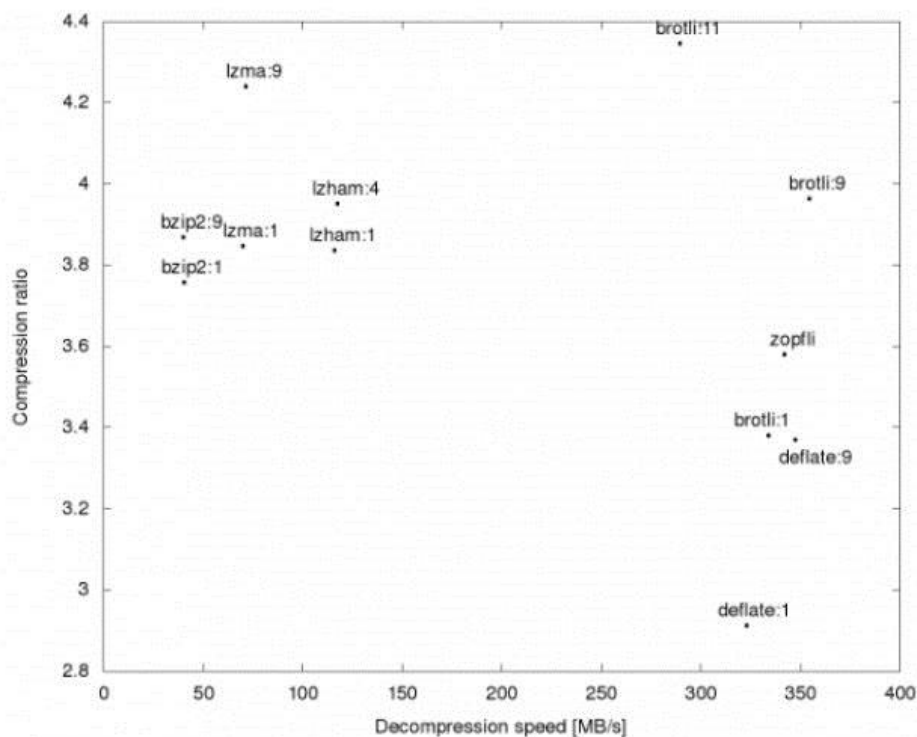
- Намаляване на packet header overhead (bandwidth savings);
- Намаляване на загубата на пакети;
- По-добро време за интерактивно реагиране;
- Намаляване на разходите за инфраструктура, повече потребители на channel bandwidth означава по-малко разходи за разполагане на инфраструктурата.

Тези предимства водят до подобро качество на QoS в мрежата и възможност операторите да подобрят своите ARPU. За потребителите, това води до по-добра QoS в мрежата и повече услуги и съдържание на връзките.

- **Използване на Brotli алгоритъм за компресия**

Brotli принадлежи към компресиращите алгоритми с общо ползване и се използва основно за минимизиране обема на данни при мрежовите връзки. Спецификите му са подадени към IETF (Internet Engineering Task Force) и сега Brotli е претендент за включване в Интернет стандартите. Новият алгоритъм се използва за компресия на шрифтовете Web Open Font Format 2.0. Оригиналният Brotli е написан на C++ и се разпространява с Apache 2.0. лиценз.

Brotli постига ниво на компресия от нивото на най-добрите съвременни методи за компресия, но изпреварва всички останали по скоростите на кодиране и декодиране.



- **Включен HTTP pipelining**

HTTP pipelining е техника, при която множество HTTP заявки се изпращат на една TCP връзка, без да се чакат съответните отговори.

Техниката беше заменена от мултиплексиране чрез HTTP / 2 [2], което се поддържа от повечето съвременни браузъри.

Внедряването на pipelining в уеб сървърите е относително проста задача - мрежовите буфери не трябва да се изчистват между заявките. По тази причина повечето съвременни уеб сървъри се справят с pipelining безпроблемно.

Внедряване в уеб браузъри - от всички основни браузъри, само в Опера имаше напълно работеща имплементация, която беше разрешена по подразбиране. Във всички други браузъри HTTP pipelining е деактивиран или не е имплементиран изобщо.

Например - Internet Explorer 11 не поддържа pipelining, Mozilla поддържа. Той обаче е деактивиран по подразбиране, за да се избегнат проблеми със сървъри с лошо управление. Когато включването в мрежата е активирано, браузърите на Mozilla използват някои

евристики, особено за да изключат pipelining за по-стари IIS сървъри. Поддръжката на H1 Pipeline бе премахната от Mozilla Firefox във Версия 54. Google Chrome преди поддържаше pipelining, но към момента е деактивиран поради бъгове и проблеми с лошо поведение на сървърите.

- HTTP/2 Server push, приоритизиращ специфични компоненти, изграждащи страниците

HTTP / 2 Server Push позволява на HTTP / 2-съвместим сървър да изпраща ресурси към HTTP / 2 съвместим клиент, преди клиентът да ги поиска. Това е, в по-голямата си част, техника за ефективност, която може да бъде полезна при предварително зареждане на ресурси.

С HTTP / 2 Push, сървърът може да поеме инициативата, като има правила, които задействат съдържанието, което трябва да бъде изпратено, дори преди да бъде поискано, което намалява възможността за загуба на bandwidth, ако изпратените до клиента подадени ресурси останат неизползвани.

- Публичните потребителски интерфейси ще поддържат адаптивен избор на TLS cipher suites според вида на процесорната архитектура на клиентското устройство - AES-GCM за x86 работни станции и преносими компютри (с налични AES-NI CPU разширения), и ChaCha20/Poly1305 за мобилни устройства (основно базирани на ARM процесори);
- Ако клиентският браузър/клиент не поддържа HTTP/2, ще бъде предвиден fall-back механизъм към HTTP/1.1. Тази възможност трябва да може лесно да се реконфигурира в бъдеще и да отпадне, когато браузърите/клиентите, не поддържащи HTTP/2, станат незначителен процент.

6.2.6.4. Качество и сигурност на програмните продукти и приложенията

Ще се спазват добрите практики на софтуерната разработка – над 60% документиране на изходния код, използване на среда за непрекъсната интеграция (Continuous Integration) , възможност за компилиране и пакетиране на продукта с една команда, възможност за инсталиране на нова версия на сървъра с една команда, система за управление на зависимостите (Dependency Management);

Публичните модули, които ще предоставят информация и електронни услуги в Интернет, ще отговарят на актуалните уеб стандарти за визуализиране на съдържание.

6.2.6.5. Информационна сигурност и интегритет на данните

Системата ще притежава високо ниво на сигурност и защита на данните при експлоатация и ще гарантира надеждно съхраняване и архивиране на информацията.

Тя ще позволява дефиниране на профили за достъп до информацията за потребителите. Всеки профил ще контролира достъпа до функции от системата и в същото време достъпа до определена част от данните в базата.

Системата ще реализира следния комплект от мерки и решения за сигурност и защита на данните:

- Осигуряване на цялостност на данните при многопотребителски режим на работа;
- Разрешаване на достъп до системата само след успешно идентифициране на потребителя. Идентифицирането ще се извърши чрез проверка на уникално потребителско име и парола, различни от тези на операционната система, на системата за управление на релационна база данни и на интегрирания OLAP-продукт. Паролите ще бъдат съхранявани в шифриран вид и ще съдържат не по-малко от 6 символа. Потребителите ще могат да променят паролите си самостоятелно;
- Регистриране на всички неуспешни опити за достъп до системата;
- Реализиране на механизъм за дефиниране на групи от потребители и за предоставяне и контролиране на права за достъп на ниво потребителски групи (роли);
- Реализиране на механизъм за предоставяне и контролиране на права за достъп до ресурсите на системата на ниво отделен служител в зависимост от конкретните задължения и отговорности на служителя;
- Регистриране на служебна информация за всички действия на потребители, касаещи регистриране, промяна и/или изтриване на данни;
- Съхраняване на история на промените в микроданните;

Системата няма не предявява специални изисквания за антивирусна защита извън приетите в НСИ решения.

Системата ще работи върху надеждно защитен уеб сървър, достъпът до който се осъществява по защитен протокол (HTTPS).

Системата ще осигурява следните нива на защита на достъпа до ресурсите:

- Няма да се допуска съхранението на пароли на администратори, на вътрешни и външни потребители и на акаунти за достъп на системи (ако такива се използват) в явен вид. Всички пароли трябва да бъдат защитени с подходящи сигурни алгоритми (напр. BCrypt, PBKDF2, scrypt (RFC 7914) за съхранение на пароли и където е възможно, да се използва и прозрачно криптиране на данните в СУБД със сертификати (transparent data-at-rest encryption);
- Ще бъде предвидена система за ежедневно създаване на резервни копия на данните, които да се съхраняват извън инфраструктурата на системата по ред, определен с наредбата по чл. 43, ал. 2 от НОИИСРЕАУ;
- Няма да бъде допуснато използването на Self-Signed сертификати за публични услуги;
- Всички уебстраници (вътрешни и публично достъпни в Интернет) ще бъдат достъпни единствено и само през протокол HTTPS. Криптирането ще се базира на

сигурен сертификат с валидирана идентичност (Verified Identity), позволяващ задължително прилагане на TLS 1.2, който е издаден от удостоверятелен орган, разпознаван от най-често използваните браузъри (Microsoft Internet Explorer, Google Chrome, Mozilla Firefox). Идентификацията се осъществява двустранно по протокол TLS (Transport Layer Security – Сигурност на транспортния слой), версия 1.2 или по-висока, дефиниран в Препоръка RFC 5246, приета от IETF (The Internet Engineering Task Force – Целева група за Интернет инженеринг) през август 2008 г. Ежегодното преиздаване и подновяване на сертификата трябва да бъде включено като разходи и дейности в гаранционната поддръжка за целия срок на поддръжката;

- Ще бъдат извършени тестове за сигурност на всички уебстраници, като минимум чрез автоматизирани средства на SSL Labs за изпитване на сървърна сигурност (<https://www.ssllabs.com/ssltest/>). За нуждите на автентикация с КЕП трябва да се предвиди имплементирането на обратен прокси сървър (Reverse Proxy) с балансиране на натоварването, който да препраща клиентските сертификати към вътрешните приложни сървъри с нестандартно поле (дефинирано в процеса на разработка на Системата) в HTTP Header-a. Схемата за проксиране на заявките трябва да бъде защитена от Spoofing;
- Като временна мярка за съвместимост настройките на уеб сървърите и Reverse Proxy сървърите ще бъдат балансирани така, че Системата да позволява използване и на клиентски браузъри, поддържащи по-стария протокол TLS 1.1. Това изключение от общите изисквания за информационна сигурност не се прилага за достъпа на служебни потребители от държавната администрация и доставчици на обществени услуги, които имат служебен достъп до ресурси на Системата;
- При разгръщането на всички уеб услуги (Web Services) ще се използва единствено протокол HTTPS със задължително прилагане на минимум TLS 1.2;
- Програмният код ще включва методи за автоматична санитизация на въвежданите данни и потребителски действия за защита от злонамерени атаки, като минимум SQL инжекции, XSS атаки и други познати методи за атаки, и ще отговаря, където е необходимо, на Наредбата за оперативна съвместимост и информационна сигурност;
- При проектирането и разработката на компонентите на Системата и при подготовката и разгръщането на средите ще се спазват последните актуални препоръки на OWASP (Open Web Application Security Project);
- Ще бъде изграден модул за проследимост на действия и събития в Системата. За всяко действие (добавяне, изтриване, модификация, четене) ще регистрира и съдържа информация със следните атрибути:
 - Уникален номер;
 - Точно време на възникване на събитието;
 - Вид (номенклатура от идентификатори за вид събитие);
 - Данни за информационна система, където е възникнало събитието;
 - Име или идентификатор на компонент в информационната система, регистрирал събитието;
 - Приоритет;

- Описание на събитието;
- Данни за събитието.
- Астрономическото време за удостоверяване настъпването на факти с правно или техническо значение се отчита с точност до година, дата, час, минута, секунда и при технологична необходимост - милисекунда, изписани в съответствие със стандарта БДС ISO 8601:2006;
- Астрономическото време за удостоверяване настъпването на факти с правно значение и на такива, за които се изисква противопоставимост, ще бъде удостоверявано с електронен времеви печат по смисъла на Глава III, Раздел 6 от Регламент ЕС 910/2014. Ще бъде реализирана функционалност за получаване на точно астрономическо време, отговарящо на горните условия, и от доставчик на доверителни услуги или от държавен орган, осигуряващ такава услуга, отговаряща на изискванията на RFC 3161;
- Ще бъдат проведени тестове за проникване (penetration tests), с които да се идентифицират и коригират слаби места в сигурността на Системата.

6.2.6.6. Използваемост

6.2.6.6.1. Общи изисквания за използваемост и достъпност

Технологичната платформа на база на която ще бъде разработена системата отговаря на утвърдените стандарти за достъпност (accessibility) и ползваемост (usability). При дизайна на потребителския интерфейс се използва модула за теми на ASP.Net MVC (theming engine), което ще позволи да се имплементира унифицирана рамка, или основна страница (Master page). Всички други страници от системата взимат общата рамка на основната страница, където е зададена основната HTML структура и референциите към CSS файловете и другите ресурсни файлове. По този начин се гарантира, че всяка една от страниците ще има сходен дизайн и ще бъде валидирана като документ с ниво „strict” според препоръките на W3C/WAI.

По отношение на потребителския интерфейс, софтуерното решение ще се придържа към следните принципи:

- При проектирането и разработката на софтуерните компоненти и потребителските интерфейси ще се спазват стандартите за достъпност на потребителския интерфейс за хора с увреждания WCAG 2.0, съответстващ на ISO/IEC 40500:2012;
- Всички ресурси ще са достъпни чрез GET заявка на уникален адрес (URL). Не се допуска използване на POST за достигане до формуляр за подаване на заявление, за генериране на справка и други;
- Функционалностите на потребителския интерфейс на Системата ще бъдат независими от използваните от потребителите интернет браузъри и устройства, при условие че последните са версии в период на поддръжка от съответните производители. Ще бъде осигурена възможност за ползване на публичните модули

на приложимите услуги през мобилни устройства – таблети и смарт-телефони, чрез оптимизация на потребителските интерфейси за мобилни устройства (Responsive Design);

- При разработката на уеб базирани страници ще се използват и да се реализира поддръжка на:
 - Стандартните семантични елементи на HTML5 (HTML Semantic Elements);
 - JSON-LD 1.0 (<http://www.w3.org/TR/json-ld/>);
 - Open Graph Protocol (<http://ogp.me>) за осигуряване на поддръжка за качествено споделяне на ресурси в социални мрежи и мобилни приложения;
- В екранните форми на Системата ще се използват потребителски бутони с унифициран размер и лесни за разбиране текстове в еднакъв стил.
- Всички текстови елементи от потребителския интерфейс ще бъдат визуализирани с шрифтове, които са подходящи за изобразяване на екран и които осигуряват максимална съвместимост и еднакво възпроизвеждане под различни клиентски операционни системи и браузъри. Няма да бъдат използвани серифни шрифтове (Serif).
- Полета, опции от менюта и командни бутони, които не са разрешени конкретно за ролята на влезлия в системата потребител, няма да са достъпни за този потребител. Това няма да отменя необходимостта от ограничаване на достъпа до бизнес логиката на приложението чрез декларативен или програмен подход.
- Всяка екранна форма ще има наименование, което да се изписва в горната част на екранната форма. Наименованията ще подсказват на потребителя какво е предназначението на формата.
- Всички търсения ще са нечувствителни към малки и главни букви.
- Полетата за пароли задължително ще различават малки и главни букви.
- Полетата за потребителски имена ще позволяват използване на имейл адреси като потребителско име, включително да допускат всички символи, регламентирани в RFC 1123, за наименоуването на хостове;
- Главните и малките букви на въвежданите данни се запазват непроменени, не се допуска Системата да променя капитализацията на данните, въведени от потребителите.
- Системата ще позволява въвеждане на данни, съдържащи както български, така и символи на официалните езици на ЕС.
- Наименованията на полетата ще са достатъчно описателни, като максимално се доближават до характера на съдържащите се в тях данни.
- Системата ще поддържа прекъсване на потребителски сесии при липса на активност. Времето ще може да се променя от администратора на системата без промяна в изходния код. Настройките за време за прекъсване на неактивни сесии ще включват и възможността администраторите да дефинират стилизирана страница с информативно съобщение, към която Системата да пренасочва автоматично браузърите на потребителите в случай на прекъсната сесия;

- Дългите списъци с резултати ще се разделят на номерирани страници с подходящи навигационни елементи за преминаване към предишна, следваща, първа и последна страница, към конкретна страница. Навигационните елементи трябва да са логически обособени и свързани със съответния списък и да се визуализират в началото и в края на HTML контейнера, съдържащ списъка;
- За големите йерархически категоризации ще се предвиди възможност за навигация по нива или чрез отложено зареждане (lazy load).

6.2.6.6.2. Интернационализация

Смарт Системс т, вземайки предвид изискванията посочени от Възложителя в техническата спецификация, предвижда:

- Системата ще може да съхранява и едновременно да визуализира данни и съдържание, което е въведено/генерирано на различни езици, а именно – двуезична на български и на английски език;
- Всички софтуерни компоненти на Системата, използваните софтуерни библиотеки и развойни комплекти, приложните сървъри и сървърите за управление на бази данни, елементите от потребителския интерфейс, програмно-приложните интерфейси, уеб услугите и др. ще се поддържат стандартно и ще са конфигурирани изрично за спазване на минимум Unicode 5.2 стандарт при съхранението и обработката на текстови данни, съответно трябва да се използва само UTF-8 кодиране на текстовите данни.
- При визуализация на числа ще се използва разделител за хиляди (интервал).
- При визуализация на дати и точно време в елементи от потребителския интерфейс в генерирани справки или в електронни документи всички формати за дата и час ще са съобразени с избория от потребителя език/локация в настройките на неговия профил:
 - За България стандартният формат е „DD.MM.YYYY HH:MM:SS”, като наличието на време към датата е в зависимост от вида на визуализираната информация и бизнес-смисъла от показването на точно време;
 - Системата трябва да поддържа и всички формати съгласно ISO БДС 8601:2006.

6.2.6.6.3. Изисквания за използваемост на потребителския интерфейс

Потребителският интерфейс ще бъде изграден при спазване на следните изисквания:

- Ще бъде реализирана възможност за добавяне и редактиране от страна на администраторите на Системата, без да са необходими промени в изходния код, на контекстна помощна информация;
- Ще бъде разработена контекстна помощна информация за всички процеси, екрани и електронни форми, включително ясни указания за попълване и разяснения за

особеностите при попълване на различните групи полета или на отделни полета;

- Контекстната помощна информация, указанията към потребителите и информативните текстове за всяка електронна административна услуга няма да съдържа акроними, имена и референции към нормативни документи, които са въведени като обикновен текст (plain-text). Всички акроними, референции към нормативни документи, формуляри, изисквания и др. ще бъдат разработени като хипервръзки към съответните актуални версии на нормативни документи и/или към съответния речник/списък с акроними и термини;
- Достъпът на потребителя до контекстната помощна информация ще бъде реализиран по унифициран и консистентен начин чрез подходящи навигационни елементи, като например чрез подходящо разположени микро бутони с икони, разположени до/пред/след етикета на съответния елемент, за който се отнася контекстната помощ, или чрез обработка на "Mouse Hover/Mouse Over" събития;
- Потребителският интерфейс ще бъде достъпен за хора с увреждания съгласно изискванията на чл. 48, ал. 5 от ЗОП.

6.2.7. Системен журнал

Изгражданото решение ще осигурява проследимост на действията на всеки потребител (одит), както и версия на предишното състояние на данните, които той е променил в резултат на своите действия - системен журнал.

Атрибутите, които ще се запазват при всеки запис, ще включват като минимум следните данни:

- дата/час на действието;
- модул на системата, в който се извършва действието;
- действие;
- обект, над който е извършено действието;
- допълнителна информация;
- IP адрес и браузър на потребителя.

Размерът на журнала на потребителските действия нараства по време на работа на всяка система, което налага по-различното му третиране от гледна точка на организация на базата данни:

- по време на работа на Системата потребителският журнал ще се записва в специализиран компонент, който поддържа много бързо добавяне на записи; този подход се налага, за да не се забавя излишно работата на Системата;
- специална фоновая задача ще акумулира записаните данни и да ги организира в отделна специално предвидена за целта база данни, отделна от работната база данни

на Системата;

- данните в специализираната база данни ще се архивират и изчистват, като в специализираната база данни ще бъде достъпна информация за не повече от 2 месеца назад; при необходимост от информация за предишен период администраторът на Системата първо ще възстанови архивните данни;
- ще бъде предоставен достъп до системния журнал на органите на реда чрез потребителски или програмен интерфейс; за достъпа ще се изисква електронна идентификация.

6.2.8. Дизайн на бази данни и взаимодействие с тях

При реализацията на решението, ще се използва стандартна релационна база данни, Microsoft SQL Server 2016 x64 Standard Edition, с която разполага НСИ..

При използване на база данни (релационна) ще бъдат следвани добрите практики за дизайн и взаимодействие с базата данни, в т.ч.:

- дизайнът на схемата на базата данни ще бъде с максимално ниво на нормализация, освен ако това не би навредило сериозно на производителността;
- базата данни ще може да оперира в клъстер; в определени случаи ще бъде използван т.нар. sharding;
- имената на таблиците и колоните ще следват унифицирана конвенция;
- ще бъдат създадени индекси по определени колони, така че да се оптимизират най-често използваните заявки; създаването на индекс ще е мотивирано и подкрепено със замервания;
- връзките между таблици ще бъдат дефинирани чрез foreign key;
- периодически ще бъде правен анализ на заявките, включително чрез EXPLAIN (при SQL бази данни), и ще бъдат предприети мерки за оптимизиране на бавните такива;
- задължително ще се използват транзакции, като нивото на изолация ще бъде мотивирано в предадената документация;
- при операции върху много записи (batch) ще се избягват дългопродължаващи транзакции;
- заявките ще бъдат ограничени в броя записи, които връщат;
- при използване на ORM или на друг слой на абстракция между приложението и базата данни, ще се минимизира броят на излишните заявки (т.нар. n+1 selects проблем);
- при използване на нерелационна база данни ще се използват по-бързи и компактни протоколи за комуникация, ако такива са достъпни.

7. ПОДХОД ЗА ИЗПЪЛНЕНИЕ НА ИЗИСКВАНИЯТА КЪМ ПОРЪЧКАТА

7.1. Предложение за извършване на дейностите по анализ и проектиране

При извършване на дейностите по анализ и проектиране Сمارт Системс т ще следва Методологията за усъвършенстване на работните процеси за предоставяне на административни услуги и Наръчника за прилагане на методологията, приета с Решение № 578 на Министерския съвет от 30 септември 2013 г.;

7.1.1. Методология за анализ на процесите и моделиране

Реализацията на софтуерни системи с нарастваща сложност изисква прилагането на общи стандарти и ефективни методологии за проектиране и разработка. От появата си през 1997г. до сега Унифицираният език за моделиране – UML™ – се превърна фактически в стандартен „графичен език за визуализиране, специфициране, конструиране и документиране на елементите на една софтуерно-интензивна система“. Сред предимствата на един UML подход са:

- Плавен преход между отделните етапи при разработката на софтуерния продукт – от формулирането на изискванията до крайната реализация, чрез използване на междинни модели за анализ и проектиране на архитектурата и поведението на системата, които позволяват всяко изискване да бъде адресирано на подходящото ниво на абстракция;
- Възможност за ефикасно разширяване на съществуващи модели с цел включване на допълнителни функционални изисквания в една бързо променяща се бизнес среда – проектът може да расте запазвайки ефективна и ясна архитектура;
- Стандартен графичен език за изразяване и дискусия на идеи, изисквания и проектантски решения, който улеснява комуникацията, особено при паралелна разработка от различни екипи (например при outsourcing на разработката);
- Един UML-базиран процес за разработка позволява да се настрои количеството и качеството на проектната документация според изискванията на конкретния проект (време, обем, бюджет, нужда от бъдещо разширяване).

UML моделирането ще бъде основно средство в процеса на анализа и проектирането на системата. С негова помощ ще се постигнат няколко основни цели:

- формално описание на единната система и нейните компоненти (подсистеми);
- описание на начина на имплементация на работата на функционалността;
- провеждане на експерименти с модела (симулация) и база за взимане на решения за оптимизиране на системата.

UML осигурява независимост на създадените модели от техническата реализация в информационните и комуникационни технологии. Прилагането на стандарта в този проект ще бъде гъвкаво и съобразено със спецификата на анализирания процес, по време на проектиране и създаване на моделите ще се запазва смисловото съдържание на моделираните обекти.

UML дефинира правила за изграждане на различни типове диаграми, които служат за графично представяне на различни аспекти на софтуерната система. Общият UML модел на който ще бъде разработен за софтуерната система се разглежда като изграден от два взаимно допълващи се модела:

Структурен модел (Structural model), който показва структурата на системата и подсистемите, използвайки обекти, атрибути, операции и връзки;

Функционален модел (Functional model), наричан още динамичен модел (Dynamic model), който показва функционалността на системата, обособена в конкретните и модули и измененията на системата във времето.

Всеки елемент от модела се описва с дадена UML диаграма - частично графично представяне на един от двата системни модела. В нотацията на UML съществуват 14 официални типа диаграми:

- **Клас диаграма** (Class diagram) – описва класове от обекти и връзките между тях;
- **Обектна диаграма** (Object diagram) – представя цялостен или частичен изглед на даден обект на системата;
- **Диаграма на пакети** (Package diagram) – представя йерархична структура на модул или компонент на системата;
- **Диаграми на компоненти** (Component diagram) – описва структурата и връзките между компонентите;
- **Диаграма на съставна структура** (Composite structure diagram) – представя декомпозиция на клас и комуникацията му с другите класове;
- **Диаграма за разгръщане/диаграма на внедряването** (Deployment diagram) – представя връзката между софтуерната и хардуерната реализация на системата;
- **Профилна диаграма** (Profile Diagram) – използва се за описване на класовете и пакетите;
- **Случай на употреба** (Use Case Diagram) – представя начина на взаимодействие между потребител и системата, употребява се за специфициране на функционалните изисквания към разработваната софтуерна система и изготвяне на тестове;
- **Диаграма на дейност** (Activity diagram) – описва последователни или паралелни контролни потоци;
- **Диаграма на състоянията** (State diagram) – описва промяната на състоянието на определен обект или система по време на неговия жизнен цикъл в резултат на възникващи събития;

- **Диаграма на последователност** (Sequence diagram) – служи за представяне на взаимодействието между обектите, като се набляга върху последователността на дейностите;
- **Диаграма на комуникация** (Communication diagram) – описва взаимодействията между обектите, като особено внимание се отделя на връзките между тях;
- **Диаграма за преглед на взаимодействието** (Interaction overview diagram) – представлява комбинация от диаграми на последователност и комуникационни диаграми;
- **Времева диаграма** (Timing Diagram) – представя взаимодействие между обекти, като се набляга върху синхронизацията на обектите.

BPMN

Business Process Model and Notation (BPMN) е другият стандарт, който ще се използва за визуално моделиране на бизнес процеси под формата на диаграма – **Business Process Diagram (BPD)**. BPMN надгражда Unified Modeling Language (UML), като целта е да се разшири техническата насоченост на описанието в UML, създавайки универсален език за описание на бизнес процеси, който да е еднакво четим и удобен за бизнес потребителите (мениджъри, бизнес анализатори и др.) и в същото време да дава възможност за описание на сложна семантика на бизнес процесите.

Поради спецификата на проекта и широкия набор от функционалности, която трябва да се реализира, използването на неформалния метод би довело до създаване на описания с твърде голям обем, чието систематично интерпретиране би било сериозно затруднение. Подобна сложна система е много по-подходяща за описание чрез диаграми и модели на данните, отколкото с текстово описание на изискванията. По тази причина най-подходящият подход към дейността е използването на полуформален подход, чрез комбинирането на документ със спецификация на изискванията, с модел от UML диаграми, представящи систематично изискванията към архитектурата на системата и отделните потребителски случаи (use cases).

7.1.1.1. Методи, стъпки и инструменти за реализация на дейностите по анализ и проектиране

Дейностите свързани с анализа и проектирането ще бъдат от ключово значение за успеха на разработката, тъй като Системата съдържа множество обвързани компоненти, които представляват сериозно предизвикателство пред използването на гъвкави подходи от екстремното програмиране базирани на прототипиране и на изграждане на системата итеративно на малки „парчета“. Потребителите на системата трудно могат да оценят изградения модел, ако той не е до голяма степен завършен, както и ако не е имплементирана съответна аналитична функционалност, която да покаже как потребителя консумира информацията. По този начин възможността за изграждане чрез итеративно подобряване на прототипи (throw away prototypes), не би била разумна от гледна точка на нужния ресурс и време за проектиране, разработка, зареждане с данни и тестване на тези прототипи. В тази

връзка, избраният подход акцентира върху правилното специфициране на изискванията и максимално близка работа с екипа на Възложителя през целия процес на проектиране и разработка, така че да се постигне голяма степен на завършеност на реализацията преди пилотното използване, а след това да се премине към отстраняване на грешки и доразвиване на основната функционалност.

Предлаганата методология за разработка на процеса – RUP, поставя спецификацията на потребителските случаи (use case) в основата на процеса на анализ на изискванията и проектиране, разработване и тестване на системата. Модела на потребителските случаи е гръбнака на формалното описание на системата, той се изготвя в процеса на спецификация на изискванията, като дори може да се каже че процеса на спецификация на изискванията бива иницииран чрез процеса на спецификация на потребителските случаи, тъй като дефинирането на use case сценарии е естествен начин за извличане на изискванията към системата. **Инструментите** за изпълнение на дейностите по моделиране ще бъдат **Microsoft Visio** – софтуерен продукт за създаване на диаграми и проектиране, тясно интегрирана с **Microsoft Visual Studio** – средата за разработка на .Net приложения. Microsoft Visio е утвърден продукт, който съдържа нужната функционалност за генериране на всички видове UML диаграми нужни за анализа и проектирането на системата – Use Case диаграми, клас диаграми, описания на моделите на архитектурата и т.н..

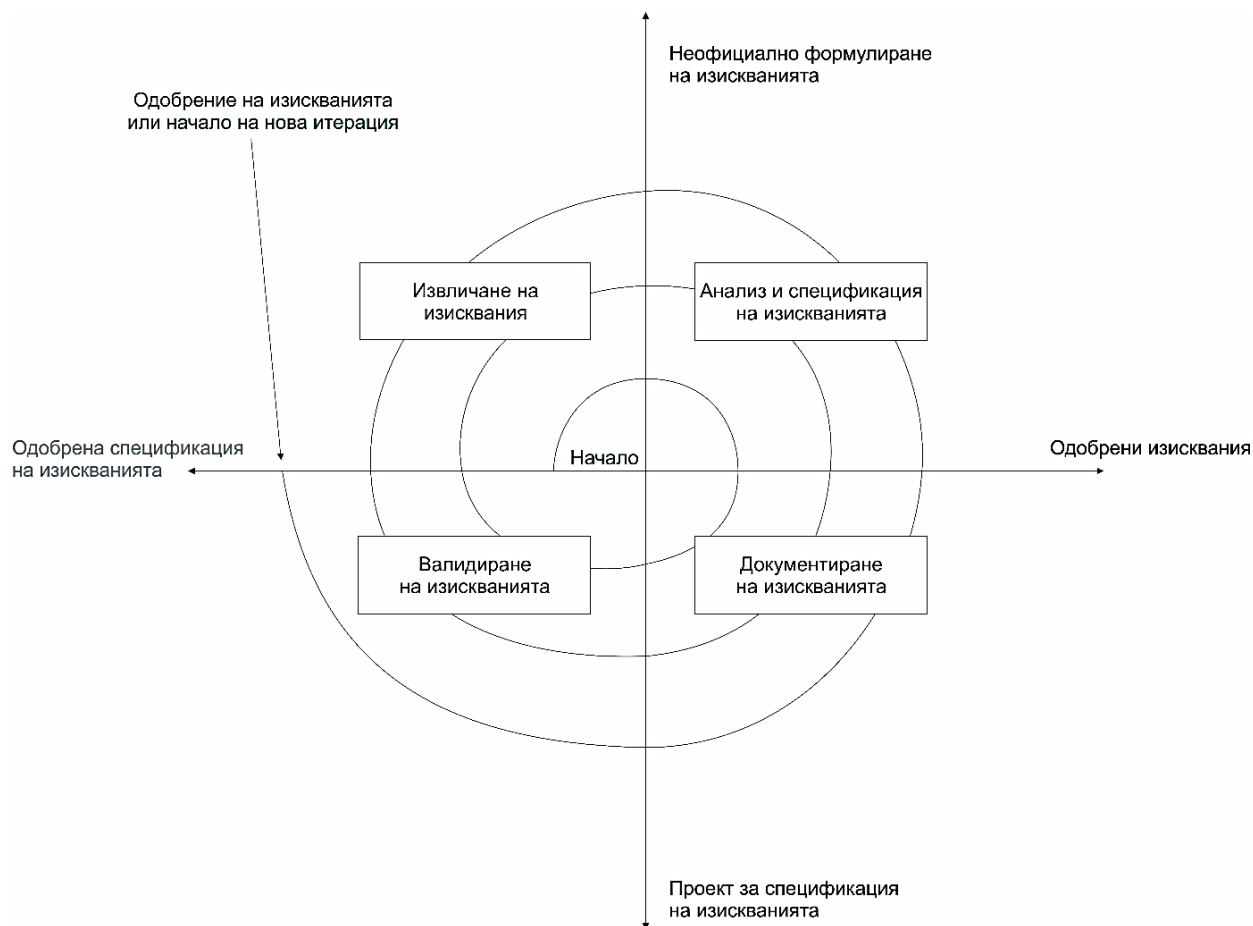
Първоначалния анализ на наличните източници на информация и срещи с екипа на Възложителят ще определят обхвата на модела на потребителските случаи, който следва да бъде правилно дефиниран в спецификацията за да се постигне пълното постигане на целите на проекта. Продуктът от тази дейност ще бъде изготвянето на спецификацията на изискванията както и на use cases, които да послужат като основа за проектирането и тестването на системата. Работата по спецификация на изискванията ще бъде надградена в процеса на изготвяне на модела на потребителските случаи (use case model), а по-късно и в етапа на проектиране, където ще се специфицира подхода за реализация на специфицираните изисквания.

При изпълнение на дейността по спецификация на изискванията ще бъде възприет итеративен подход към подготовката на детайлната спецификация. При този подход екипа на Сمارт Системс ще извършва в итерации основните стъпки от процеса на формализиране на изискванията:

- Извличане на изискванията на база на изследване на идентифицираните източници на информация, както и на база на резултатите от проведените срещи и дискусии с екипа на Възложителя;
- Анализ и дефиниране на изискванията, в процес на обсъждане с Възложителя;
- Документиране на изискванията в проект на Спецификация на изискванията;
- Валидиране на изискванията, документираните в проекта на Спецификацията на изискванията, завършващо с обратна връзка от екипа на Възложителя.

На база на обратната връзка от Възложителя, в края на итерацията от стъпки, екипът от анализа или инициира наново процеса на извличане и анализ на изискванията, или се приема процеса на специфициране на изискванията за приключен.

Итеративния подход и съответстващите му дейности и резултати са описани графично на следната диаграма:



Фигура 15 Диаграма на итеративния подход и съответстващите му дейности и резултати

7.1.2. Метод за спецификация на потребителските случаи

Потребителските случаи (use case) се смятат за стандартна нотация при изготвянето на изискванията в обектно-ориентирано моделиране. Потребителските случаи на употреба могат да подобрят комуникацията между заинтересованите страни, експерти по бизнес анализ, разработчици и други специалисти участващи в разработването на системата.

Разглеждат се три различни нива на абстракция на изискванията (бизнес, потребителски и системни) и това как потребителските случаи могат да бъдат определящ фактор при разработването на детайлните изисквания за разработване на системата.

✓ Класификация на изискванията

Изискванията към Система като тази, определят неговото функционално поведение, както и типовете информация, до която трябва да се осигури достъп, как тя се трансформира и организира.

Определянето на изискванията дават възможност на участниците в проекта да уточнят ясно целта, насоката и да изяснят очакванията от информационните цели. Тъй като

често участниците – бъдещи потребителите на системата дават доста обобщени бизнес изисквания, а екипът разработващ системата има нужда от точни, ясни и недвусмислено очертани изисквания се налага допълнително изясняване, с цел те да се трансформират в детайлни, пълни и подлежащи на тестване спецификации, на базата на които да бъде разработена система максимално отговаряща на нуждите на Възложителя.

✓ **Бизнес изисквания**

Бизнес изискванията представят цели на високо равнище, които следва да се реализират чрез внедряването на системата. Те са описани в Техническата спецификация, която описва визията и обхвата на проекта. Бизнес изискванията идентифицират първичните ползи, които системата ще донесе на Поръчителя и потребителите. Те представляват най-високото ниво абстракция във веригата на изискванията. Те описват целите, възможностите и основните потребителски типове, които дават обобщена представа, за това как би следвало да функционира системата.

✓ **Потребителски изисквания**

Потребителските изисквания описват задачите и действията, които даден потребител трябва да може да извършва с помощта на системата. Тези изисквания трябва да се уточнят с хората, които на практика ще използват системата. Тези потребители са в състояние да опишат не само задачите, които е нужно да извършват в системата, но и нефункционалните ѝ характеристики, които са важни за доброто приемане и лесната работа със системата.

Потребителските изисквания следва да са в синхрон с бизнес изискванията. Те могат да бъдат уточнени най-пълноценно посредством описанието на потребителските сценарии. Фокусират се върху конкретните нужди при използване на системата и съответно са доста по-значими от традиционният подход за извличане на изисквания, посредством директно питане на потребителите какво биха искали да прави системата.

✓ **Детайлни системни изисквания**

Те представляват системните изисквания на много детайлно ниво, което подпомага пълното, детайлно специфициране на изискванията, позволяваща използването на така специфицираните изисквания при изграждане на компонентите от разработчиците.

Тези изисквания трябва да съответстват на потребителските и бизнес изисквания и да съдържат точни и ясни критерии, които да позволяват верификация.

Функционалните изисквания определят функционалностите, които разработващият екип следва да реализира при изграждането на системата, така че системата да удовлетворява бизнес изискванията. Бизнес изискванията улавят очакваното поведение на системата, което може да бъде изразено чрез услуги, задачи и функции, от които системата се нуждае, за да отговори на целта, заложена в проекта.

Информационните изисквания дефинират информационните нужди на Поръчителя и заинтересованите страни. Те описват типовете информация и данни, които системата следва да предостави, или до които да даде достъп. Те специфицират предоставените данни, посредством качеството, което те следва да имат, източникът от който идват, как би следвало да бъдат обработвани, как следва да бъдат комбинирани за анализ и кои аналитични методи следва да бъдат използвани.

Другите изисквания, извън функционалните и информационните, могат да бъдат специфицирани, за да се опишат допълнителни аспекти на системата, като изисквания към интерфейса и средата (законова, културна, политическа).

✓ **Качествени характеристики на изискванията**

Добавянето на качествени характеристика към описанието на функционалните, информационните и други изисквания, позволява оценяването им в определени измерения, които са важни или за потребителите или за екипа, разработващ системата.

Характеристиките на изискванията са свойства или качества, които системата трябва да притежава. Те могат да съдържат стандарти, регулации и условия, с които системата трябва да бъде съобразена: описания на външния интерфейс, изисквания за производителността, дизайна и ограничения при изпълнението; качествени характеристики.

✓ **Идентифициране на изисквания посредством потребителски случаи (use cases)**

Анализаторите от дълги години използват потребителските случаи на употреба (use cases), за да опишат начините, по които потребителя взаимодейства със системата. Целта е да се извлекат релевантни и точни изисквания. Използването на потребителски случаи възниква с напредъка на обектно-ориентирания подход към програмирането, подходът който е заложен в основата на подхода за проектиране и разработка на предлаганото решение.

✓ **Модел на потребителските случаи и процес на изготвяне на детайлната спецификация**

Описанието на потребителските случаи често не дава достатъчно пълна информация на екипа разработващ системата за функционалностите, които те следва да разработят, както и за това каква и коя информация следва да бъде достъпна. Събирането на всички изисквания от потребителските случаи може да доведе до формирането на твърде тромави и комплексни потребителски случаи. От друга страна, спирането на развитието на изискванията по време на стадия на събиране на потребителските изисквания ще доведе до много пропуски в информацията, от която екипът разработващ системата ще има нужда по време на стадия на конструирането ѝ. За да се понижи тази несигурност всеки потребителски случай трябва да бъде разработен заедно с пълните си системни изисквания.

Всеки потребителски случай на употреба води до определянето на няколко системни изисквания, които ще позволят на специфицирането на дадена функционалност в разработеното приложение и съответно няколко потребителски случая могат да формират спецификацията на една и съща функционалност или компонент.

Специфицирането на функционалностите, компоненти и изискванията към тяхното качество може да бъде описано чрез детайлни системни изисквания, които могат да бъдат асоциирани с потребителските случаи по няколко начина.

Подходът, който ще бъде избран зависи от това дали екипът, разработващ системата трябва да създаде дизайна, конструкцията и тестовете, базирайки се на потребителски случаи, на детайлни системни изисквания или на комбинация от двете. Избирайки подходящия подход е важно, да се избягва дублирането на информация от различни източници, което би направило спазването на изискванията много по-трудно.

✓ **Итеративно и инкрементално разработване на детайлната спецификация**

Екипът на Сمارт Системс предлага итеративния и инкрементален подход, който е в основата на RUP да бъде приложен по същия начин към дейността по изготвяне на детайлната спецификация. Итеративният подход към изготвянето на детайлната спецификация включва пресяването на потребителските случаи посредством няколко итерации. Итеративните стъпки към събирането на изисквания от потребителски случаи е строго индивидуално според ситуацията. Предварително няма как да се определи точен брой итерации, които ще бъдат необходими, за да се покрият изискванията във всички ситуации. Но може да се каже, че спецификацията на итеративните изисквания винаги преминава през същите логически стъпки във всички ситуации.

7.1.2.1. Подход към осигуряване на качеството на продукта от дейностите по спецификация на системния проект

При осигуряването на качеството на извършените дейности ще бъде използван като референтен модел препоръките на IEEE - Recommendation for SRS. Използвайки IEEE Standard 830, Recommended Practice for Software Requirements Specifications, ръководството на екипа ще съблюдава правилата и критериите за качество при изготвяне на спецификацията. Целта на въпросните действия по осигуряване на качеството е да се гарантира, че изготвения документ ще предложи стабилна и ясна рамка за проектирането и изграждането на технологичната реализация. Така специфицираните изисквания ще отговарят на изискванията за качество на спецификацията на изискванията:

✓ **Точност**

Този атрибут не е формално проверим в процеса на качествен контрол, но итеративния модел на валидиране на изисквания позволява на крайния потребител – екипът на Възложителя да провери документираните изисквания, по отношение на това как те представят описаната визия за изискванията към решението.

✓ **Недвусмисленост**

Недвусмислеността на изискванията ще бъде проверена чрез преглед на изискванията от трети лица, тъй като основните проблеми при изготвянето на изискванията идват в различното тълкуване на едно и също изискване от разработчиците и бъдещите потребители.

✓ **Пълнота**

Тук ще се приложи сравнение със стандартите за съдържание на спецификацията, така че да се сравни дали всички задължителни елементи присъстват и са опасни в нужната пълнота. При итеративния процес на подготовка, всички непълни или отсъстващите елементи ще бъдат ясно индикирани и коментирани, така че в крайния вариант да не се допусне одобрение без наличие на пълния набор от елементи.

✓ **Не противоречивост**

Чрез използване на речник на термините и дефиниции за основните предположения ще се избегне конфликт между отделни части от документа за спецификация на

изискванията, както и между спецификацията и други документи свързани с изпълнението на проекта.

✓ **Приоритизирани по важност**

Процеса на приоритизиране на изискванията следва да бъде извършен от екипа на Възложителя, така че от една страна да се обсъди аргументацията на приоритетите, така че екипът по изпълнение да възприеме логиката и да научи съответните ключови предположения, а от друга страна да се насочи вниманието и на екипа по разработка.

✓ **Проверими**

Работата по проверка на този индикатор за качество ще бъде изключително полезна в рамките на подготовката на тестовия план. Проверката се изразява в дефиниране на ясен критерий за проверка на всяко изискване.

✓ **Променими**

С цел лесната итеративна работа по подобрене на изискванията те ще бъдат документирани в съответствие със структурата на спецификация при спазване на ясни правила за идентифициране и рефериране на отделните изисквания, осигуряващи че всяко изискване се включва еднократно в спецификацията, а се референцира от другите части на документа, ако има зависимост с друго изискване. По този начин промените в документа не нарушават неговата консистентност.

✓ **Проследими**

С цел проследимост на развитието на изискванията, всяко изискване ще бъде идентифицирано с уникален номер, като се поддържа и информация за неговата идентификация в по-ранни версии на документа.

7.1.3. Методика за проектиране

Проектирането на компонентите ще бъде извършено от екипа на Сمارт Системс съобразно съвременни, доказани и препоръчвани практики в областта на софтуерния дизайн, които са доказали своята приложимост при разработката на комплексни софтуерни системи.

В съответствие с RUP, процесът на проектиране ще бъде извършен като итеративен процес на дефиниране и подобряване на архитектурата, на дизайна на компонентите, интерфейсите и останалите характеристики на компонентите. В основата на дейността ще се състои в това изискванията към системата да се анализират с цел да се изработи визия за нейната вътрешна структура, която след това да се развие в цялостна архитектура, която да послужи за основа на разработката. Така разработената архитектура ще описва как системата се декомпозира и организира на отделни компоненти и интерфейсите между тези компоненти. След това детайлно се проектира и функционалността и интерфейса на всеки от компонентите, така че да се постигне необходимото ниво на детайлност на това описание до степен, която да позволи на разработчиците да извършат самото изграждане на компонента. Описаният подход, избран от екипа на Смарт Системс, се характеризира от следните основни дейности:

- Дизайн на софтуерната архитектура (дизайн на високо ниво) – описва се структурата

и организацията на системата на високо ниво. Идентифицират се отделните компоненти, от които тя се състои.

- **Дизайн на компонентите/модулите** – всеки компонент/модул се описва с ниво на детайлност, което е достатъчно за неговото конструиране.

Резултатът от дейността по проектиране ще бъде логическия дизайн и организация е съвкупност от модели и артефакти, които съдържат/описват взетите важни решения в процеса на тяхното изготвяне.

Прилагането на гъвкава и модулна архитектура, структурирана по начин, който едновременно осигурява консистентност на данните и максимална защита от грешки, а от друга страна позволява лесно надграждане и разширяване на системата, без този процес да бъде съпътстван от регресивни грешки поради скрита свързаност на модулите, е изключително сложна задача. За постигне на нейните цели следва да се спазват основните принципи на обектно ориентирания дизайн.

7.1.3.1. Принципи за дизайн

В основата на добрият софтуерен дизайн е спазването на определени принципи за дизайн, които са доказали своята ефективност и ефикасност при множество успешни софтуерни системи:

- ✓ **Абстракция** – абстракцията може да бъде определена като процес на “забравяне” на информация, така че дадено множество от различни по своята същност неща да могат да бъдат третираны като еднакви. Абстракцията по същество представлява опростяване на проблемната област, като цената която се плаща е загуба на информация;
- ✓ **Свързаност и кохезия** – свързаността/обвързаността (coupling) дефинира силата на връзката между отделните модули, докато кохезията (cohesion) дефинира доколко вътрешните елементи на даден модул са взаимно (логически) свързани и служат за изпълнението на неговите функции. При наличие на висока свързаност между модулите, ако трябва да се направи изменение в даден модул е твърде вероятно това да наложи промени и в свързаните модули. Обратно при ниска свързаност промените обикновено се ограничават до модула, който реализира изменената функционалност. Високата кохезия предполага, че логически свързаните елементи са групирани заедно. Това спомага да се намали сложността, тъй като се елиминират сложните последователности от взаимодействия между различните модули. В идеалния случай модулите, от които се състои дадена софтуерна система трябва да имат ниска свързаност и висока кохезия.
- ✓ **Декомпозиция и модулност** – големите и сложни софтуерни системи е добре да се декомпозират на отделни под-системи и модули, като целта е да се разположат различните функционалности и отговорности в различни компоненти. По този начин се постига разделяне на отделни компоненти, чиято сложност е управляема.
- ✓ **Енкапсулация** – енкапсулацията (encapsulation/information hiding) представлява

групиране и пакетиране на елементите и вътрешните детайли на дадена абстракция, правейки ги недостъпни отвън. Целта е да се постигне потенциал за промяна: вътрешните механизми на компонента могат да бъдат променяни и подобрявани без това да оказва влияние на останалите компоненти. Обикновено се енкапсулират онези аспекти на компонента, които има вероятност да бъдат променяни в бъдеще.

- ✓ **Разделяне на интерфейсите от имплементацията** – принципът е свързан с принципа на скриване на информацията. Интеракцията с дадена абстракция се осъществява само чрез строго дефинирани публично достъпни интерфейси, вътрешната структура остава скрита.
- ✓ **Достатъчност, пълнота и простота на дизайна** – постигането на достатъчност, пълнота и простота означава, че софтуерния компонент обхваща всички важни характеристики на абстракцията, която реализира и нищо повече.

7.1.3.2. Шаблони за дизайн (Design patterns)

При обектно ориентирания дизайн стремежът е системата да се планира като съвкупност от взаимодействащи обекти. Всеки обект представя даден елемент от системата, която се моделира и се характеризира със своето състояние и поведение. Обектът представлява абстракция групираща в себе си данни и процедури. Интеракцията с обекта става посредством добре дефиниран интерфейс на обекта. Ключова роля при обектно ориентирания дизайн играят понятията наследяване и полиморфизъм.

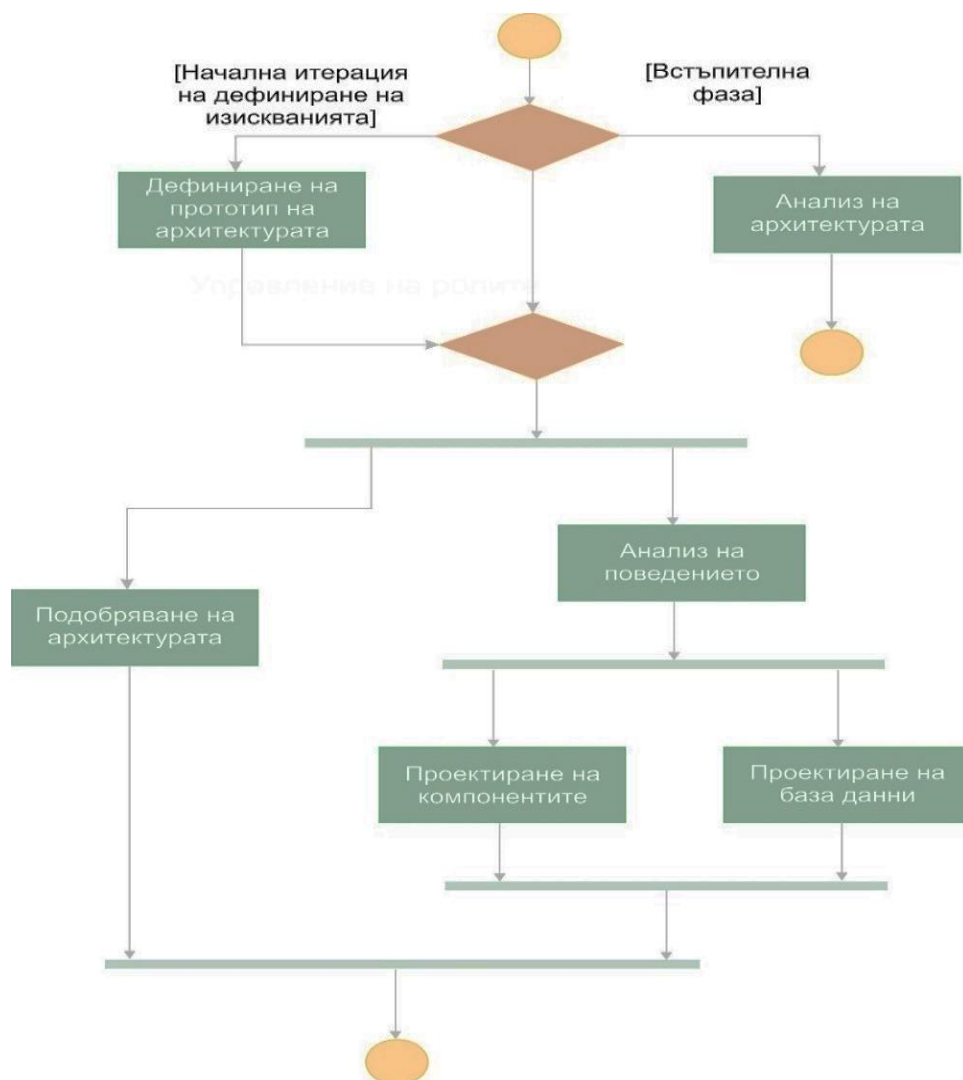
Подходът при обектно ориентирания дизайн е проблемния домейн да се декомпозира на отделни обекти следвайки принципите на дизайн от общото към частното и свързване на частите съобразно техните отговорности. Обектно-ориентирания анализ и дизайн по същество е процес на последователни действия на опростяване т.е. оперирайки с проблемния домейн в него се „инжектират“ структури, които го декомпонират и опростяват. Структурите, които служат за декомпозиция и опростяване, представляват шаблони за дизайн (design patterns), които са доказали своята ефективност при много различни ситуации. Докато различните архитектурни стилове за софтуерен дизайн представляват шаблони от високо ниво (определят макроархитектурата), съществуват шаблони за дизайн описващи детайлите на по-ниско, локално ниво (определят микроархитектурата). Такива видове са: Creational patterns (builder, factory, prototype, singleton), Structural patterns (adapter, bridge, composite, decorator, façade, flyweight, proxy), Behavioral patterns (command, interpreter, iterator, mediator, memento, observer, state, strategy, template, visitor).

7.1.3.3. Приложение на итеративния подход в обектно – ориентирано проектиране

Основните методи в процеса на разработка на софтуер, в предлаганата от Сمارт Системс методология – процес за разработка на софтуер RUP, се базират изключително на итеративния обектно-ориентиран подход на анализ и дизайн на системата. В началните итерации артефактите ще бъдат предимно документи, описващи изискванията и съдържащи аналитични и технически UML модели. Следващите итерации ще създадат софтуерни

версии, които реализират специфицираните изисквания. Последните итерации ще се съсредоточат върху тестове, отстраняване на програмни грешки, внедряване на софтуера и евентуално очертаване на бъдещото ѝ развитие.

На диаграмата е показана архитектурата на процесите за обектно ориентиран анализ и проектиране:



Фигура 16 Архитектурата на процесите за обектно ориентиран анализ и проектиране

В началото фокусът се насочва върху изграждането на първоначална архитектура на компонентите (Define a Candidate Architecture), за да може да бъде дадена отправна точка за главната работа по анализ. В случай, че вече съществува архитектура, която е била изготвена на предишни итерации, в предишни проекти или зададена като изискване, фокусът на работата се измества върху нейното актуализиране (Refine the Architecture). Следващата стъпка е да се анализира очакваното поведение на системата и да се специфицират конкретните елементи, които изграждат това поведение (Analyze Behavior).

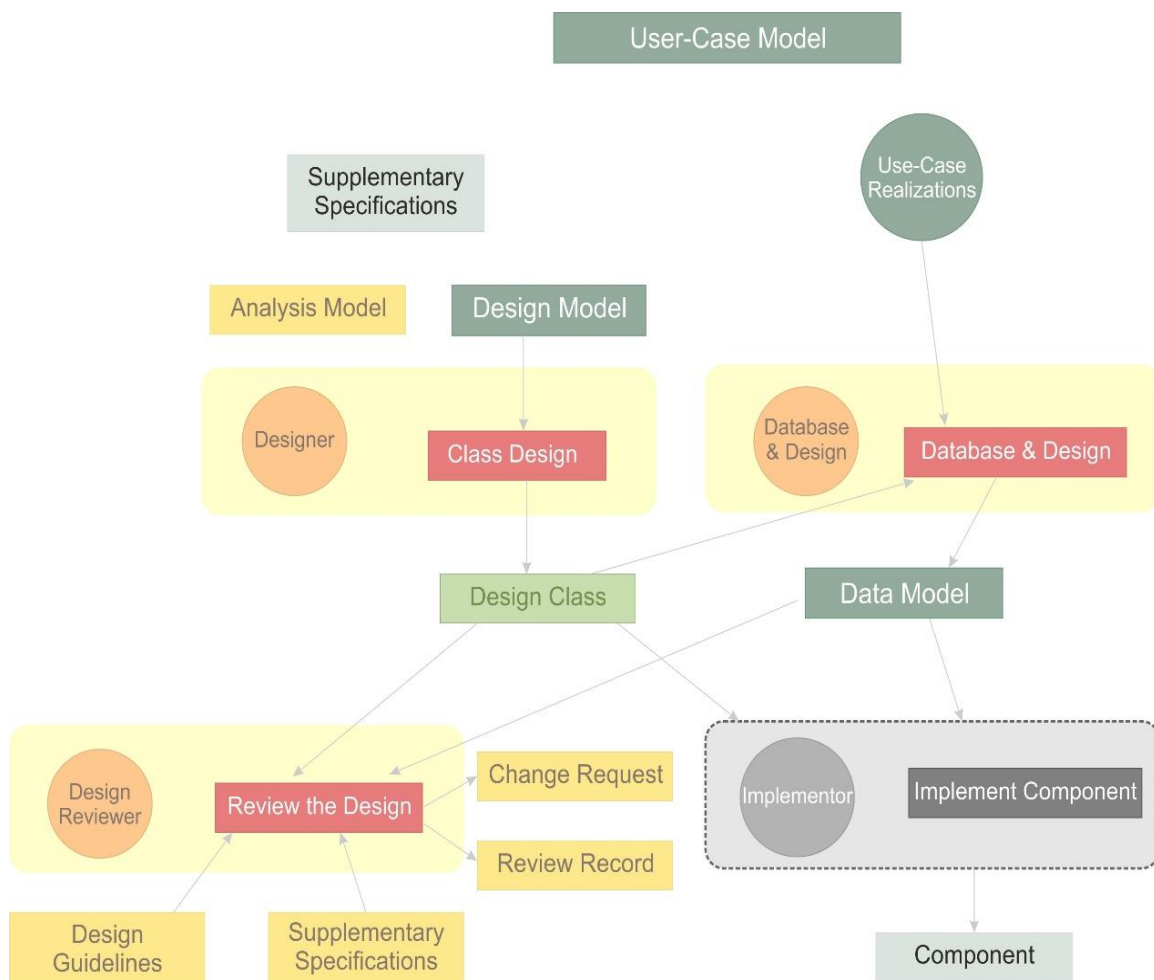
След като първоначалните елементи бъдат идентифицирани, следва да бъдат допълнително детайлизирани в етапите на дизайн на компонентите и базата данни. Дизайна на компоненти (Design Components) изготвя множество компоненти, които доставят изискваното поведение на системата. Паралелно с това, анализ на начините за съхранението им в базата данни бива извършен в дизайна на базата данни (Design the Database).

Реализацията на тази концепция и постигането на обектно-ориентиран анализ и дизайн на системата се извършва на базата на готовите дефинирани изисквания под формата на Модел на потребителските случаи (Use Cases), от които се изгражда тяхната реализация под формата на Клас диаграми (Class diagrams) и Диаграми на последователността и взаимодействието (Sequence diagrams и Collaboration diagrams, съответно). Това е имплементационния модел на системата, който включва още Дизайн модел и Модел на данните.

Дизайн моделът е обектен модел, който описва реализацията на потребителски случаи и служи за извеждане на Модела на имплементацията и неговия програмен код.

Моделът на данните е подмножество на имплементационния модел, което описва логическия и физически вид на постоянните (персистентни) данни в системата. Той включва и видовете поведения в базата данни, например записани процедури, активатори, ограничения и др.

На диаграмата е показана процесната архитектура за дизайн на компонентите:



Фигура 17 Диаграма на процесната архитектура за дизайн на компонентите

Използването на модели ще позволи лесна промяна и бъдещо разширение на съществуващите услуги и добавяне на нови електронни услуги.

7.1.3.4. Методика за проектиране на потребителските интерфейси

Потребителският интерфейс ще бъде изграден чрез използването на стандартите HTML 5 и CSS 3. Всички уеб страници ще бъдат изградени така, че структурата и съдържанието на страницата (HTML) да бъдат в максимална степен отделени от тяхното визуално представяне (CSS). Това ще позволи при необходимост лесната модификация на визуалните елементи.

При изграждането на потребителския интерфейс ще бъде използван подхода Responsive Web Design (RWD), който ще осигури гъвкавост и възможност за оптимално представяне на данните върху различните браузъри. За реализирането на богат, динамичен и удобен за работа потребителски интерфейс, който не изисква често презареждане на съдържанието от сървъра, презентационния слой ще бъде реализиран като Single Web

Application. Ще бъдат използвани библиотеки за динамично уеб съдържание jQuery (<http://jquery.com/>), KnockoutJS (<http://knockoutjs.com>) или AngularJS (<http://angularjs.org/>). Конкретната библиотека, която ще бъде използвана за динамично уеб съдържание ще бъде избрана съобразно конкретните изисквания към потребителския интерфейс дефинирани в резултат на анализа.

При проектирането на потребителския интерфейс ще бъде използван подхода залегнал в стандарта Human centered design processes for interactive systems (ISO 13407) и препоръките на W3C в Web Content Accessibility Guidelines 2.0 (ISO/IEC 40500:2012).

Потребителският интерфейс ще спазва следните добри практики:

- ✓ Интерфейсът предоставян от всеки модул ще бъде ориентиран към реализираните от него функции. Когато потребителят има нужда от достъп до информация в друг модул ще бъде предвидена възможност за пряк достъп до нея без да е необходима навигация през менютата на системата
- ✓ Опциите в менютата и достъпната функционалност в интерфейса на системата ще зависят от ролята и правата за достъп на потребителя, екранните форми няма да бъдат утежнени с излишни функции
- ✓ Потребителят ще бъде информиран от системата за успеха или неуспеха на изпълняваните от него операции
- ✓ Логически свързаните полета ще бъдат групирани;
- ✓ Всяко поле ще бъде означено с етикет подсказващ неговата функция;
- ✓ Полетата ще бъдат разположени на екрана, така че да се ограничи дължината на вертикалния скрол. Хоризонтален скрол ще се допуска по изключение;
- ✓ Всяка форма извършваща промяна в данните ще притежава бутон “Отказ”, при натискането на който ще се отменя действието
- ✓ Съобщенията за грешка във въвежданите данни ще съдържат ясна информация за вида на грешката и очаквания формат на данните.

7.1.4. Инструменти за описание на аналитичните дейности

Business Process Model & Notation (BPMN) представлява стандартизиран и съвременен вариант за дефиниране и анализ на бизнес процеси. BPMN се състои от стандартна нотация, която е интуитивна за разбиране от различните роли в дадена организация – мениджмънт, бизнес анализатори, технически специалисти и програмисти. Основната цел на BPMN е да подпомага управлението на бизнес процеси (business process management), като предоставя универсален език за комуникация между ролите създаващи дизайн на бизнес процеси и техните изпълнители. След преценка на добрите практики във функционалното моделиране и на спецификите на настоящия проект, включително традициите в агенцията, измежду богатото разнообразие от изразни средства в UML, „Смарт Системс 2010“ ЕООД избра следните диаграми, които ще се използват за функционалните модели на новата система в своята техническа и експлоатационна документация:

UML диаграми

Целта на този точка е да се направи въведение в унифицирания език за моделиране UML. UML е стандартен (ISO/IEC 19501:2005), език за моделиране с общо приложение в областта на софтуерното инженерство, създаден от Grady Booch, Ivar Jacobson и James Rumbaugh в Rational Software през 1990те, последващо придобит от Object Management Group (OMG) в 1997, където се управлява и досега.

От 2000 година UML се приема от International Organization for Standardization (ISO) като индустриален стандарт за моделиране на софтуерни системи.

Диаграма на дейностите

„Смарт Системс 2010“ ЕООД ще използва диаграми на дейностите, за да моделира последователността от действия, които описват определена функционалност на системата.

Диаграмата на дейности е техника за описване на процедурна логика, бизнес процес и работен поток.

Тя е като блок-схема, но поддържа паралелно поведение. Може да има разклонения както при блоксхемите. Показва логиката на поведение. Състоянията са дейности. Дейност е изпълнението на задача, като това може да бъде физическа дейност или изпълнение на код. Диаграмата на дейности показва последователност от действия.

Диаграмата на дейностите е по-удобна за описание на процеси на бизнес ниво, поради две причини:

- Показва всички възможни развития на процеса (успешен и не успешен завършек)
- Описва логиката на процесите, без да описва средствата или градивните елементи.
- Начало

Всяка диаграма на дейности има едно начало, от което започва последователността от действия.



Фигура 18 Начало на диаграма за дейности

- Край

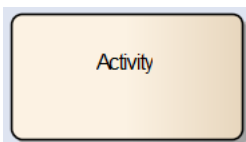
Всяка диаграма на дейности има един, край където завършва последователността от действия.



Фигура 19 Край на диаграма за дейности

- Действие

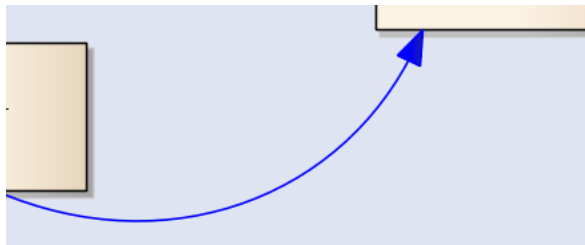
Действията са свързани помежду си с преходи.



Фигура 20 Действие в диаграма за дейности

- Преходи

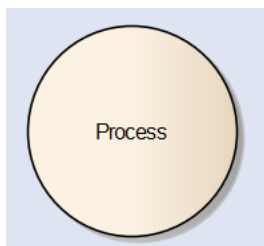
Преходите се означават със стрелки и показват посоката от предишно към следващо действие. Понякога съдържат текст.



Фигура 21 Преходи в диаграма за дейности

- Процес

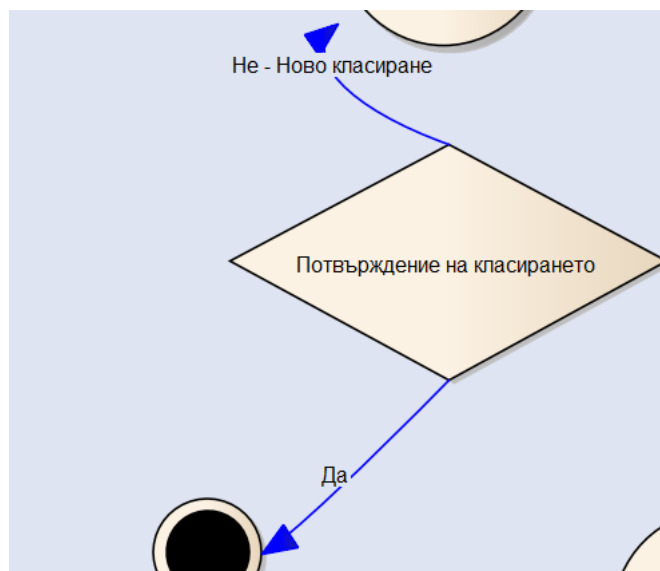
Процесите се визуализират с кръг.



Фигура 22 Процес в диаграма за дейности

- Процес за проверка

Осъществява се проверка за валидация на формата. Ако е валидна се изпраща формата и с това приключва процеса, а ако не е валидна се връща процеса в попълване на формата.



Фигура 23 Процес за проверка

- Обекти



Фигура 24 Обекти в диаграма за дейности

Use Case Diagram (Диаграма на потребителските случаи)

В процеса на проектиране на софтуерния продукт, диаграмата на потребителските случаи е първата диаграма, която се създава от проектантите, когато се започне проект. Тази диаграма позволява да се опишат на най-високо ниво целите на потребителя, които системата трябва да изпълнява. Тези цели не е необходимо да са задачи или действия, а може да са по общи изисквания към функционалността на системата. С други думи това е техника за определяне на функционалните изисквания на една система. Те описват типичните взаимодействия между потребителите и системата, предоставят описание на начина, по който тя се използва.

Примерите за използване (Use-cases) са основани на сценарий техника в UML, която идентифицира актьорите в едно взаимодействие и която описва самото взаимодействие. Множество от примери на използване би трябвало да опише всички взаимодействия със системата.

Сценариите са примери от реалния живот, как системата може да се използва.

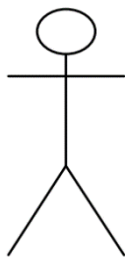
Те трябва да включват:

- Описание на началната ситуация
- Описание на нормалния поток от събития
- Описание на това, което не е както трябва

- Информация за други дейности, извършвани в същия момент
- Описание на състоянието, в което сценарият завършва

Компоненти при Use case диаграми

- Actor (Актьор) – роля, която един потребител играе по отношение на системата. Може да бъде клиент, представител по поддръжка на клиенти, мениджър продажби, продуктов аналитик. Актьорите изпълняват случаите на употреба. Един актьор може да изпълнява много случаи на употреба и един случай на употреба може да се изпълнява от много актьори. Актьора клиент може да представлява множество хора. Един човек може да играе ролята на няколко актьора. Актьора може да не е човек, а друга компютърна система, ако системата извършва услуга за нея.



Actor

- Associations – плътна връзка без стрелка. Associations са връзка между actors и use cases и означава, че актьора осъществява потребителския случай.

«uses»



- Случай на употреба



7.2. Методики за извършване на дейностите по разработка на системата

7.2.1. Методология за извършване на дейностите по разработка на системата

Предлаганата методология за разработка се основава на принципите на RUP, като за източник за логиката на реализацията се базираме на софтуерната спецификация на

изискванията и софтуерния дизайн на компонентите специфициран в системния проект, както и на изготвените в рамките на изпълнението му план за тестване и тестовите сценарии за контролиране на качеството на софтуера, част от паралелния нему процес по контрол на качеството на разработката.

Цялостната производствена процедура обхваща няколко паралелни процеса и може да бъде обобщена в следните стъпки:

- ✓ Стъпка 1 - Разработване на софтуерните модули (units);
- ✓ Стъпка 2 - Разработване на план за тестване и тестовите сценарии;
- ✓ Стъпка 3 - Разработка на модулните тестове (unit tests)
- ✓ Стъпка 4 - Проверка (тестване) на софтуерните модули;
- ✓ Стъпка 5 - Интегриране (build) на софтуерните модули;
- ✓ Стъпка 6 - Разработване и провеждане на интеграционни тестове (integration tests);
- ✓ Стъпка 7 - Зареждане на системата с начални данни;
- ✓ Стъпка 8 - Разработване и провеждане на системни тестове;
- ✓ Стъпка 9 - Произвеждане на крайния продукт и документация.

7.2.1.1. Методика за разработване на модела на данни

Предложената методика за разработване на модела на данните в системата обхваща всички етапи от жизнения цикъл на проектирането на базата данни, като отчита спецификите на отделните приложения с които трябва да се интегрира системата. Една от основните специфики на приложенията, която ще бъде взета под внимание е тяхната архитектура – някои от приложенията с които се интегрира системата са разпределени, а други са централизирани, в допълнение всяко от тях е реализирано на базата на различни технологии и платформи.

Основните етапи от жизнения цикъл на разработването на модела на данни са следните:

- Проучване и анализ на изискванията
- Изграждане на логически модел
- Проектиране на базата данни

По-надолу ще развием в детайли всеки един от тези етапи.

✓ Проучване и анализ на изискванията

Основната цел на този етап от разработването на модела данни е да се определят потребителските изисквания към базите данни. За целта е необходимо да се идентифицират компонентите и приложенията, взаимодействащи си със системата. Това са всички разработени приложения, които ще бъдат интегрирани със системата, както и възможните нови такива.

На този етап трябва ясно да се определят нуждите на потребителите, изискванията към системата и ограниченията върху решението.

Изследването на системата е особено важен етап в процеса на цялата разработка. Допуснатите пропуски на този етап рефлектират върху пълнотата на проекта. Най-общо е необходимо да се проведе проучване, както бе описано в предходната точка в следните насоки:

- Функционирането на съществуващата система и несъвършенствата в нея;
- Икономическата и организационна среда на системата и възможните промени в бъдеще;
- Проблемите, слабите места и ограниченията на системата, както и породилите ги причини;
- Основните процеси и начина на тяхното реализиране;
- Типовете данни и техния обем;
- Справките и отчетите, необходими за функциониране на системата;
- Потребителските изисквания и възможностите за тяхната реализация.

По време на проучването и анализа е важно да се осмислят взаимовръзките между отделните компоненти на системата. Указаните области на проучване не се изследват последователно. Те са дадени в този ред за по-голяма прегледност. При изследването е важно да се постигнат крайните резултати, т.е. да се опишат основните потребителски изисквания и да не се пропуснат някои важни детайли.

✓ Изграждане на логически модел

Когато се проектира една база данни, е необходимо да се вземат решения по отношение на създаването на най-удобния модел на дадена система от реалния свят. При създаване на модела на данните се извършва организиране на изискванията на системата в едно логическо представяне на базата данни. Моделът на данните се състои от обекти и техните атрибути и ограничения, дефиниции на релации между обектите и ограничения върху тези релации, следователно разработването на логически модел на данните се осъществява чрез определяне на обектите, техните атрибути и ограничения и техните релации. Създаденият по този начин модел позволява да се идентифицират съответните таблици, които трябва да бъдат създадени, колоните, съдържащи се в тях, релациите между таблиците.

- Определяне на обектите и техните колони – когато се определят обектите, е необходимо, от изискванията на системата да се дефинират основните логически подразделения на информацията. Докато се разглеждат изискванията на системата, се дефинират основните обекти и събития. В резултат на това се добавят таблици към дизайна на базата данни, които им съответстват.

След като са дефинирани всички таблици, които могат да бъдат определени в този момент, трябва да се дефинират колоните на тези таблици. Тази информация се взема директно от изискванията на системата, в които е определено какви данни трябва да се поддържат за обектите и събитията.

- ✓ **Определяне на релациите между обектите** – по време на процеса на дефиниране на релациите между таблиците може да се открие, че е необходимо да се промени създадения до този момент дизайн. Започва се с избиране на една от основните таблици и определяне на обектите, които имат релации към тази таблица.

След като се установят релациите между таблиците, трябва да се дефинира типа на тези релации. Всяка линия се маркира в двата си края или с цифрата 1, или със символа „n“. Цифрата 1 се отнася до страната едно на релацията, а символът „n“ се отнася до страната много на релацията. За да се определи типа на релацията, се разглеждат данните, които съдържат таблиците и видовете взаимоотношения между тях. В един нормализиран проект на базата данни обаче релациите много към много трябва да се модифицират, като се добави една свързваща таблица и между нея и във всяка първоначална таблица да се създадат релации едно към много. Аналогично се определят и типовете на останалите релации.

- ✓ **Определяне на ограниченията върху данните** - работните правила включват всички ограничения за една система, включително за целостта на данните и сигурността.

На този етап от процеса на изграждане на модела на данните трябва да се обърне внимание на спецификите и ограниченията върху данните. Например дадено поле може да не е задължително във формата, а да е позволено да се попълни на по-късен етап. В такъв случай полетата, които съдържат информация, която не е задължителна, да бъдат маркирани като опционални, т.е. да позволят съхраняване на стойност NULL в тези полета. От друга страна, ако разгледаме поле, което е задължително за попълване, колоната трябва задължително да има попълнена стойност и тя да съществува като стойност в колоната ID от съответната таблица.

7.2.1.2. Проектиране на базата данни

Теорията на проектиране на релационни бази данни се състои от следните основни понятия:

- Таблицы и уникалност - Първичен ключ (primary key) уникален идентификатор на ред, който представлява колона или група от колони, използвани за разграничаване всеки отделен ред от останалите редове в таблицата. Може да бъде прост или съставен. Простият ключ е създаден от една колона, съдържаща уникални стойности за всеки ред от таблицата.

Всяка таблица може да има само един първичен ключ, дори когато няколко колони или комбинации от колони съдържат уникални стойности (наречени алтернативни ключове).

Изборът на първичен ключ трябва да се основава на принципите за:

- минималност (избират се толкова колони, колкото е необходимо);
- стабилност (избират се колони, които рядко биват променяни);
- простота (от колкото е възможно по-прост тип)
- Външни ключове и домейни - Външен ключ (foreign key) колона или група от колони в дадена таблица, представляваща връзка с друга таблица посредством

нейния първичен (родителски) ключ.

Родителски ключ е ключът, към който сочи един външен ключ. Родителският ключ трябва да бъде уникален идентификатор, за да може да се определи към кой ред от таблицата на родителския ключ сочи външният ключ. Броят и типът на колоните, съставлящи външния ключ, трябва да съответства на броя и типа на колоните на родителския ключ, но могат да се използват различни имена за тях. Не е задължително стойностите на външния ключ да бъдат уникални в своята-собствена таблица. Те трябва да бъдат в същата област от допустими стойности (домейн), на която принадлежат стойностите на родителския ключ.

Домейн съвкупност от стойности, които са допустими за дадена колона.

- Релационни връзки - Релация (relationship) се нарича връзка между таблици, която е базирана на първичен ключ от едната таблица и външен ключ от другата таблица.

Съществуват три типа релации между таблиците: релации едно към едно; релации едно към много; релации много към много.

Релации едно към едно - две таблици А и В са свързани с релация едно към едно, ако за всеки ред от таблицата А има най-много един съответстващ ред от таблицата В, но всеки ред от таблицата В може да има точно един съответстващ ред от таблица А.

Релации едно към много - две таблици А и В са свързани с релация едно към много, ако за всеки ред от таблицата А има нула или повече съответстващи редове от таблицата В, но всеки ред от таблицата В може да има точно един съответстващ ред от таблицата А.

Релации много към много - две таблици А и В са свързани с релация много към много, ако за всеки ред от таблица А има много съответстващи редове от таблицата В и обратно. Този тип релация се създава, като се дефинира трета таблица, наречена свързваща таблица (junction table), която съдържа първичните ключове от двете таблици като външни ключове; първичният ключ в свързващата таблица е съставен от двата външни ключа.

- Правила за запазване на целостта на данните - целостта на данните е важно понятие за проектиране на базите данни. Има четири вида цялост на данните:
 - цялост на обект – едно от изискванията на проектирането на релационна
 - база данни е възможността да се разграничат различните инстанции на даден обект. Това понятие е известно като цялост на обект и се реализира чрез създаване на първичен ключ. Според това правило за цялост, колоните съставлящи първичния ключ, не могат да имат стойност NULL. Релационните бази данни поддържат специална стойност NULL, която указва неизвестните стойности (unknown);
 - цялост на област – свързана е с осигуряване на валидност на стойностите на колоните, т.е. да принадлежат на допустима област от стойности. Реализира се с определяне на типа на колоните, допускане на стойност NULL, ограничения, стойност по подразбиране, дефиниране на външен ключ;
 - цялост на връзка – запазва дефинираните отношения (релации) между таблиците, когато се въвеждат, променят или изтриват редове. Целостта на връзките гарантира, че съществува съгласуваност на стойностите на

ключовете между таблиците – първичните и външните в съответните таблици. Реализира се с дефиниране на ограничението външен ключ. Когато се реализира цялост на връзките, не се допуска да бъдат добавени редове в една таблица, която е страната “много” на релацията, ако в първичната таблица, която е страната “едно” на релацията, липсва съответен ред. Също така не се допуска да се променят стойностите на колоните на първичния ключ в една таблица, която е страната “едно” на релацията, ако в свързаната таблица има поне един съответен ред. Не се допуска да се изтриват редове от една първична таблица, ако има свързани редове в таблицата с външните ключове;

- дефинирана от потребителя цялост – дава възможност за определянето на специфични бизнес правила, които не могат да се отнесат към някоя от другите категории цялост. Реализира се чрез създаване на ограничения, съхранени процедури и тригери.

7.2.1.3. Разработване на софтуерните модули (units)

Софтуерните модули се разработват от програмистите на базата на софтуерните спецификации и дизайн, при спазване на вътрешните конвенции за кодиране.

Планът на проекта е основата за наблюдение на дейностите и за предприемане на коригиращи действия. Развитието на проекта се контролира главно чрез сравняване на реално получения продукт и усилията и средствата, вложени в изпълнението на задачата с времевата диаграма (schedule) на плана по зададените вътрешни срокове или контролни точки от структурирането на задачите. Когато изпълнението се отклонява съществено от плана, трябва да се предприемат подходящи коригиращи действия. Едно отклонение е значително, когато, ако се остави нерешено, ще възпрепятства постигането на целите на проекта. Коригиращите действия включват:

- Промяна на начина на работа;
- Добавяне на ресурси;
- Смяна на процесите;
- Преразглеждане на рисковете;
- Промяна на изискванията;
- Преразглеждане на оценките и плановете;
- Предоговаряне на споразуменията.

7.2.1.4. Процедура за управление на програмния код

Предлаганата процедура за управление на програмния код ще се прилага за всички резултати от изпълнението на настоящия проект - версиите на софтуерния продукт и свързаните документи представящи резултатите на изпълнението на дейностите по проекта. Функционалността нужна за прилагане на процедурата се осигурява от система за контрол на версиите (SVN) и от архивираща (back-up) система.

Исходният код (Source Code) разработван по проекта, може бъде публично достъпен онлайн като Софтуер с отворен код от първия ден на разработка, чрез използване на система за контрол на версиите.

Разработките по проекта ще бъдат задължително подаване към система за контрол на версиите. Освен кодът, цялата документация и отчетни материали ще бъдат качвани в хранилището.

Ще бъде предвидено използването на система за контрол на версиите и цялата информация за главното копие на хранилището, прието за оригинален и централен източник на съдържанието, да бъде достъпна публично, онлайн, в реално време.

Смарт Системс т ще поддържа еднакви версии на софтуера за продуктивната и тестовата среда. Новите функционалности ще се инсталират на продуктивния вариант на системата само след като са преминали успешно изпитанията в тестовата среда.

Смарт Системс т ще осигури достъп от минимум един акаунт до софтуерното хранилище (software repository) и последния актуален сорс код на Системата. Тези акаунти ще бъдат използвани от Възложителя за контрол на версиите, допълнително синхронизиране и архивиране на сорс кода на компонентите.

Предимствата на системата за контрол на версиите е това, че предоставя възможност за подробно разглеждане на направените промени, които са довели до създаването на нова версия, за сравняване на версиите една с друга и за връщане на по-стара версия като актуална.

В основата на всяка една система за контрол на версиите, както е и при Subversion, стои хранилището на информация (repository). При постъпване на нова версия на файл в хранилището, старата му версия се съхранява с дата и час или с други думи, съхраняват се всички версии на даден файл в хронологичен ред. Системата съхранява също и автора на всяка версия - по тази причина при свързване с хранилището всеки потребител се идентифицира с име и парола. Потребителите имат различни права - или да редактират файла (т.е. да запазват нови версии в хранилището) или само да четат данни. Тъй като повечето софтуерни продукти могат да работят само с по една версия на даден файл, а не с всички наведнъж, потребителят получава работно копие (working copy) на нужните му файлове, което съдържа изисканата от него версия от хранилището (в типичния случай - най-актуалната) и работи с него.

Системата за контрол на версиите е ценна не само за съхраняването на програмен код, в нея успешно могат да се съхраняват и документите изготвени при разработката на даден проект – файлове с последователните версии на планове, спецификации, документация на системата. Поддържането на такива версии в хранилището има предимство пред съхраняването им във файлови сървъри, защото се запазва точната верига на промени по един документ, а не само съхранението му към момента на копиране в папката, като по този начин може да се възстанови версия на документ към даден момент във времето.

Управление на конфигурацията

Управлението на конфигурацията е една от фундаменталните дейности от софтуерното инженерство. Ролята на управлението на конфигурацията е да управлява множеството от документи, обекти, софтуер и хардуер с цел получаване на краен продукт. Управлението на конфигурации и промени засяга и двете основни дейности на проекта.

Инструментите за конфигурация на управлението предлагат възможности за:

- **Контрол на версиите** – поддържа история на промените, направени на компонент и се развива във времето. Позволява също и достъп до определена версия, не просто последната;
- **Паралелна разработка** – няколко разработчика могат да правят различни промени на един и същ компонент по едно и също време. Тези промени по – късно се обединяват и възникналите конфликти се разрешават ръчно или автоматизирано.

Управлението на конфигурациите включва три основни дейности:

- Определяне елементите на конфигурацията;
- Контрол над съдържанието на елементите;
- Проследяване на статуса на конфигурацията.

Средства за контрол на версиите и хранилище с публичен достъп

Смарт Системс т предлага да инсталира в средата на Възложителя система за контрол на версиите Субвършън (Subversion SVN). SVN управлява ключовата информация за разработката на системата, процесите на поддръжка и инсталиране и база за потенциално преизползваеми артефакти. Хранилището за код на SVN може да бъде конфигурирано за предоставяне на публичен достъп до програмния код.

SVN дава възможност за:

- Поддържане версии на директориите и принадлежащите им файлове;
- Реална история на версиите – възможни са копираня и преименувания на файловете. Всеки нов файл започва със собствена чиста история, няма наследяване при преименуване;
- Атомарни предавания - при набор от модификации или всички се записват, или всички се отказват. Това позволява разработчиците да изграждат и предават промени като логически обединения и премахва проблемите от частичен запис;
- Версии на метаданните - всеки файл и директория има собствен набор от свойства, асоциирани с тях. Поддържат се и версии на тези свойства;
- Избор на нива от мрежата - SVN стандартно използва HTTP протокол. Може да се модифицира и за използване по SSH протокол;
- Консистентно управление на данните - SVN показва разликите във файловете използвайки бинарен алгоритъм, който работи еднакво както за текстови, така и за бинарни файлове. И двата типа се пазят в базата с данни в компресиран вид;
- Ефективно създаване разклонения и тагове;
- Използва механизъм на директно копиране и създаване на твърди връзки.

Управление на промените

Управлението на промени предлага процедури за контрол над исканията, които се смятат за отклонение от договорената основна линия на проекта. Тези процедури се прилагат към всички видове искания за промени, засягащи документи от планирането на проекта, потребителски изисквания или контрола на грешките.

Исканията за промени се повдигат и от двете участващи страни и са адресирани към другата страна за оценка и одобрение. С цел избягване на забавяне, което може да доведе до промени във времените граници на проекта, период от 10 работни дни от регистрацията на искането може да бъде приет за отправна точка при оценката и вземането на решение.

Всеки член на проекта и от двете страни може да повдигне искане за промяна, но то трябва да е съгласувано с отговорниците по качеството.

Основни стъпки на процедурата са:

- Попълване на форма за искане на промяна;
- Анализиране на искането;
- Оценяване на необходимите разходи;
- Прилагане на искането;
- Поддържане на история на исканията.

7.2.1.5. Инструменти за следене на прогреса на развитие на системата и разпределение на задачите

В рамките на проекта ще бъдат изградена среда за управление на задачите и следене на прогреса, интегрирана със среда за регистриране на проблеми с разработената функционалност – в периодите на тестване, а и на последващата гаранционна поддръжка. Цялостното проследяване, регистриране и управление на задачи, регистрирането на грешки, проблеми/несъответствия ще се осъществява посредством система за управление на задачи на Сمارт Системс , а именно **Redmine**. Системата ще работи в режим 24/7. Системата ще бъде разделена на два модула:

- Модул за регистриране на задачи към екипа за разработка – модула представлява проект в системата Redmine, в който ще имат достъп само участниците в екипа по реализация. Задачите в този проект се регистрират от ръководителя на екип и са регистрирани или на база специфицирана задача от плана на проекта (категория Feature), или на регистрирана грешка или проблем от тестване или експлоатация на системата (категория Bug). Във втория случай се регистрира връзката за съответстващия таск във втория модул.
- Модул за регистриране на грешки/проблеми/несъответствия – в този модул могат да се регистрират тестери от екипа на Изпълнителя, потребители и ИТ служители от страна на Възложителя, които регистрират открити грешки и несъответствия. Заявките се насочват към бизнес анализатора, който извършва първоначалния анализ

и дефинира задачи в проекта за задачи към разработчиците. Ръководителя на екипа одобрява дефинираните задачи и ги разпределя на съответен разработчик за изпълнение. След изпълнение на задачата в другия модул се отразява статус в модула за регистриране на проблеми.

Целта на системата е да се централизира информацията за всички задачи на екипа по разработка – както задачи по разработка на нова и надграждане на съществуваща функционалност, така и задачи във връзка с отстраняване на софтуерни грешки в една база данни. По този начин се осигурява бърз достъп до информацията за всички промени във версията върху която се работи, което облекчава планирането, следенето на напредъка, както и анализа на потенциалните регресионни отражения на промените.

Redmine позволява удобно да се регистрират, разпределят, наблюдават, планират и отчитат задачите. Модула за регистриране на проблеми улеснява както връзката на потребителите (представители на Възложителя) с отделите по поддръжка на Сمارт Системс , така и сътрудничеството и комуникацията между специалисти по поддръжката, анализатори и разработчици в самата организация на Сمارт Системс .

Всички лица с регистрация в системата имат възможност да проследяват развитието по отстраняване на проблема в реално време посредством уеб интерфейс.

Redmine е безплатен и с отворен код, уеб базиран инструмент за управление на проекти и проследяване на проблеми. Тя позволява на потребителите да управляват множество проекти и свързани с тях под проекти. Той разполага с уики и форуми за проекти, проследяване на времето и гъвкав контрол на достъпа, базиран на роли. Той включва календар и диаграми на Gantt, които подпомагат визуалното представяне на проектите и техните крайни срокове. Redmine се интегрира с различни системи за управление на версиите и включва браузър за хранилища и дифузор.

Някои от основните характеристики на Redmine са:

- Уеб базиран административен панел, позволяващ настройка и персонализиране на системата за проследяване на проблеми;
- Гъвкав контрол на достъпа, базиращ се на потребителски роли с възможност за определяне на техните права;
- Гъвкава система за проследяване на проблеми, с възможност за дефиниране на статуси и типове проблеми;
- Управление на новини, документи и файлове;
- Известия по имейл;
- Проследяване на времето за изпълнение на дадена задача;
- Персонализирани полета за проблеми, записи, проекти и потребители;
- Множествената поддръжка на LDAP удостоверяване;
- Многоезична поддръжка;
- Поддържат се множество бази данни - MySQL, PostgreSQL или SQLite.

The image shows the login form in Redmine. It is a yellow box with a white background. It contains a 'Login' label above a text input field. Below that is a 'Password' label above another text input field. To the right of the password field is a link for 'Lost password'. Below the password field is a checkbox labeled 'Stay logged in'. At the bottom is a 'Login' button.

Страницата за вход се използва за влизане в проекта, в който сте били активирани. Връзката за изгубена парола се показва само ако администраторът я е активирал.

Регистриране на потребител:

The image shows the 'Register' form in Redmine. It is a light gray box with a white background. It contains several fields: 'Login *', 'Password *' (with a note 'Must be at least 8 characters long.'), 'Confirmation *', 'First name *', 'Last name *', and 'Email *'. There is a checkbox for 'Hide my email address' which is checked. At the bottom is a 'Language' dropdown menu set to 'English'. A 'Submit' button is at the bottom left.

Въз основа на настройките на Redmine, потребителят ще трябва да активира профила си по имейл, да изчака администраторът да потвърди профила или да види активирането му автоматично.

- Активиране на профила по имейл - След като предостави необходимата информация на страницата за регистрация, потребителят ще получи имейл на имейла, предоставен на страницата за регистрация. С кликването върху връзката за активиране, предоставена в имейла, потребителят ще активира своя профил.

- Ръчно активиране на акаунта - След като предостави необходимата информация на страницата за регистрация, потребителят трябва да изчака одобрението на администратор. Администраторът ще активира потребителския акаунт на страницата "Администрация"> "Потребители". Когато потребителският профил е активиран, потребителят ще има право да влиза в системата.

Регистриране и проследяване на проблеми:

The screenshot shows the Redmine web interface. At the top, there's a navigation bar with links like Home, My page, Projects, Help, and a search bar. Below this is a secondary navigation bar with tabs for Overview, Download, Activity, Roadmap, Issues (selected), New Issue, News, Wiki, Forums, Repository, and Settings. The main content area is titled 'Issues' and includes a filter section with 'Status' set to 'open'. Below the filter is a table of issues. The table has columns for #, Tracker, Status, Subject, Updated, and Category. The right sidebar contains links for 'View all issues', 'Summary', and 'Custom queries'.

#	Tracker	Status	Subject	Updated	Category
27831	Feature	New	ATOM Feed for repository changes	2017-12-22 01:29	Feeds
27826	Feature	New	"Non member" should be invisible to the projects.	2017-12-22 04:58	Accounts / authentication
27825	Defect	New	Issues are automatically filtered as open only	2017-12-20 08:42	Issues filter
27823	Feature	New	Display only the tickets that arrived on the due date on the calendar	2017-12-19 14:39	Calendar
27822	Feature	New	Remove filename from attachment preview links	2017-12-24 02:17	Attachments
27821	Feature	New	add subject to spent time filter	2017-12-19 13:28	Time tracking
27807	Patch	New	Use a unique way to check/uncheck a group/fieldset with checkboxes	2017-12-16 10:08	UI
27804	Defect	New	Restriction of user visibility isn't working with internal authentication	2017-12-15 09:59	Security
27799	Patch	New	Mark default version in versions tab from project settings	2017-12-15 06:38	Project settings
27790	Patch	New	mercurial - fix double quotes in branch names	2017-12-13 10:55	SCM
27780	Defect	New	Attachment sort doesn't work with Unicode	2017-12-12 02:24	Attachments
27779	Defect	New	Context menu reverse-x doesn't work properly when window_height < wrapper_size	2017-12-11 16:49	Issues list

От избрана страница с проблеми можете да видите текущата работа, която е извършена, по отстраняване на проблема. Съобщенията се показват в хронологичен ред (за да промените реда - вижте настройката в "Моите профили"). Възможно е да цитирате други съобщения, както и да редактирате вашите.

Системата позволява създаване на връзка между свързани проблеми. Това улеснява работата на Сمارт Системс по отстраняване на проблеми, като в случай на дублиране, може да се премахнат излишните позиции. В случай, че даден проблем произтича от вече регистриран или има определен ред, в който проблемите следва да бъдат отстранени, това също може да се отбележи посредством такива връзки.

The screenshot shows a form titled 'Related issues'. It contains a dropdown menu with 'related to' selected, followed by a text input field for 'Issue #', and 'Add' and 'Cancel' buttons.

Добавяне на нов проблем

Хората могат да създадат нов проблем, когато отговарят на ролите и правата, конфигурирани от администратора на Redmine (Поля: Проследяване на емисиите> Добавяне на проблеми).

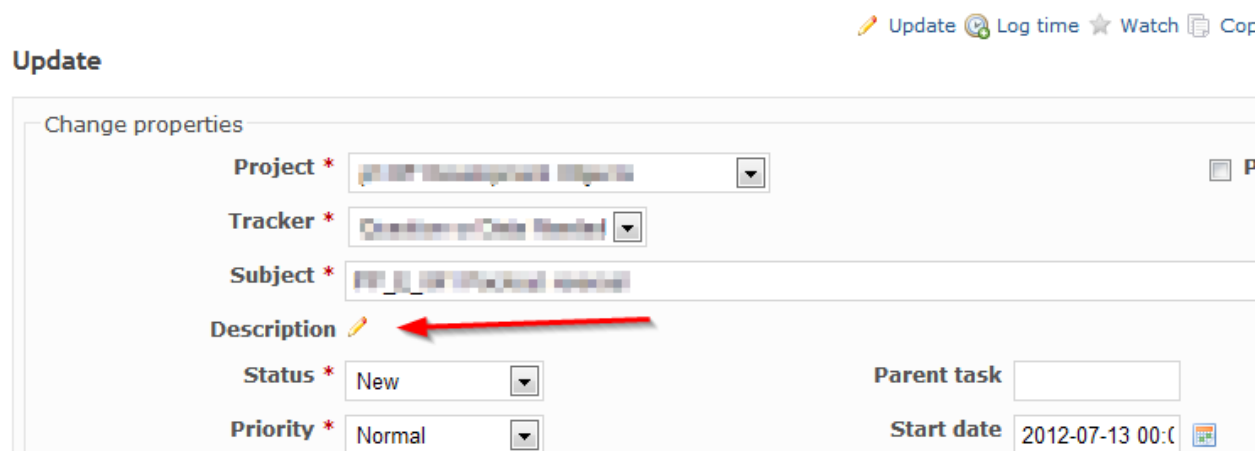
При създаването на нов проблем един от най-важните елементи е полето за проследяване, което определя естеството на проблема. По подразбиране Redmine идва с три различни тракера: бърг, функция и поддръжка.

Актуализиране на съществуващ проблем



В зависимост от правата, с които разполага дадената потребителска роля, ще бъде визуализиран пълен или ограничен набор от свойства на проблеми, които могат да се редактират.

Редактиране на предмет или описание на съществуващ проблем



7.2.1.6. Интегриране (build) на софтуерните модули

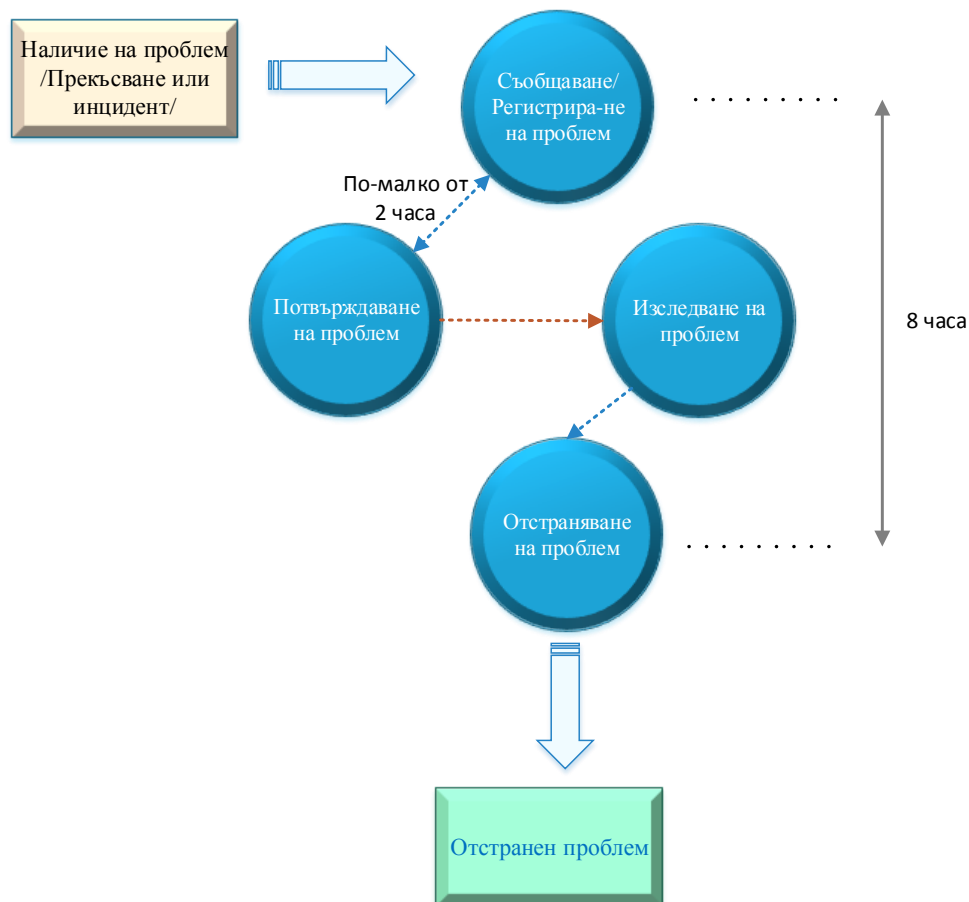
Завършените софтуерни модули се интегрират от програмистите в цялостен софтуерен продукт на базата на софтуерните спецификации и дизайн, при спазване на вътрешните конвенции за кодиране. Тук се реализират и интерфейсите за интеграция с външните за софтуерния продукт системи.

Както и при разработването на софтуерните модули и тук планът на проекта е основата за наблюдение на дейностите и за предприемане на коригиращи действия.

7.2.1.7. Процедурата за управление на софтуерни грешки/несъответствия/проблеми

Подходът, който Сمارт Системс т ще прилага при откриване на грешки/несъответствия/проблеми, е представен схематично на Фигура 18.

Гаранционна поддръжка



Фигура 25 Процедура за отстраняване на грешки/несъответствия/проблеми

Представеният подход представлява набор от последователно извършвани дейности с оглед осигуряване на безпроблемната и надеждна работа на компонентите и обхваща следните основни стъпки:

- ✓ **Съобщаване/Регистриране на грешка** (несъответствие/проблем). Проблемът се описва чрез Система за регистриране и управление на заявки за проблеми и дефекти, за да бъде разгледан и впоследствие разрешен.
- ✓ **Потвърждение на проблем** – включва дейности по преглед и потвърждаване на изпратения проблем от експерт от екипа на Сمارт Системс, като времето от съобщаването до потвърждаването ще е по-малко от 2 часа;
- ✓ **Изследване на проблем** – включва дейности по преглед и анализ на информацията по изпратения проблем – възможни причини и последствия от възникването на проблема, възможни решения и начини за отстраняване на проблема;
- ✓ **Разрешаване на проблем** – включва дейности по отстраняване на проблема, тестване на решението и възстановяване на работоспособността на услугата.

Отстраняването на проблеми ще включва следните под-дейности:

- ✓ **Възпроизвеждане и/или анализ на проблема.**
- ✓ **Намиране решение на проблема и съгласуване с бизнес и ИТ служител/и от екипа на Възложителя.**
- ✓ **Реализация на промените в средата за разработка.**
- ✓ **Тестване на промените в средата за разработка.**
- ✓ **Верификация на промените от страна на Възложителя.**

Смарт Системс предоставя на определен служител от екипа на Възложителя версия с всички корекции, извършени през дейностите на тестване и последваща гаранционната поддръжка на Source-кода върху електронен носител на информация CD/DVD, като се подписва приемо-предавателен протокол.

7.2.2. Методика за внедряване

За изпълнение на дейностите с необходимото качество, в срок и в съответствие с изискванията по проекта, при планиране ще бъдат предприети действия за определяне на:

- процедурата за внедряване, включително подготвителните действия за внедряване на системата;
- ресурсите, необходими за изпълнение, включително екипа от експерти;
- спецификация на тестовете за приемане на системата в окончателен вариант;
- процедура за изграждане на продукционната среда;
- график за изпълнение на дейностите.

Организацията на изпълнение като минимум ще включва:

- разработване на инструкцията за изпълнение на процедурата за внедряване;
- подготовка на експертите, включени в екипа за внедряване, за изпълнение на дейностите по процедурите за изграждане на продукционна среда и за внедряване;
- разработване на образците на документи за отчитане изпълнението на дейностите.

Всички дейности във фазата на планирането ще бъдат съгласувани с Възложителя.

Резултатите от дейностите по планиране внедряването на системата ще бъдат обобщени в План за внедряване, който ще бъде представен на Възложителя за одобрение – на хартия и в електронен формат. Планът за внедряване като минимум ще включва:

- Подготовка за внедряване – описание на подготвителните дейности, които трябва да бъдат извършени преди внедряване;
- Процедура за внедряване – детайлно описание на дейностите с подробни указания за

изпълнение на всяка стъпка;

- Процедура за изграждане на продукционна среда с подробно описание на действията за инсталация и конфигуриране на компонентите на системата;
- Екип за изпълнение – списък на експертите, включени в екипа за изпълнение и данни за контакт (стационарен телефон, мобилен номер, е-мейл адрес, административен адрес за кореспонденция и т. н.);
- Спецификация на тестовете за приемане на системата;
- Образец на протокол за изграждане на продукционна среда;
- Образец на протокол от проведени тестове за приемане на система, в който ще се отразяват резултатите от изпълнение на тестовете;
- Образец на протокол за внедряване за отчитане изпълнението на дейностите;
- График за изпълнение.

✓ **Изграждане на продукционна среда**

Изграждането на продукционната среда ще бъде извършено по начина, определен във фазата на планиране на внедряването и в съответствие с разработената процедурата за изграждане на продукционна среда.

✓ **Изпълнение на дейностите по внедряване**

При внедряване на компонентите ще бъдат извършени действията, определени във фазата на планиране на внедряването, в съответствие с разработената процедура за внедряване.

Дейностите ще бъдат извършени по одобрените от Възложителя План за внедряване и график за изпълнение и по предварителна оценка ще включват:

- параметризация на базата данни, включително създаване на системните класификатори, конфигуриране на системата;
- зареждане в базата данни на мигрираните данни;
- провеждане на тестове за приемане на системата, в хода които ще се извърши и представяне пред екип на Възложителя на реализираните функционалности, което ще гарантира навременно запознаване със системата.

✓ **Отчитане изпълнението на дейностите по внедряване**

Отчитане изпълнението на дейностите по внедряване ще бъде включено в междинния доклад за изпълнение на Етап 4: Внедряване, в който ще бъдат изложени поставените цели и постигнатите резултати. Ако в хода на изпълнение са възникнали проблеми, те ще бъдат документирани. Ще бъдат описани и предприетите действия за преодоляването им. Към доклада ще бъдат приложени оригинали на:

- протокол за изграждане на продукционна среда;
- протокол от изпълнението на дейностите по внедряване;

- протокол от проведените тестове за приемане на системата.

7.2.2.1. Начин на прилагане на предлагания подход за внедряване на компонентите

✓ Планиране на внедряването и организация на изпълнението

Във фазата на планиране на внедряването ще бъдат предприети следните конкретни стъпки за подготовка и изпълнение на дейностите:

- ще бъде разработена процедурата за внедряване с указания за изпълнение на всяка стъпка;
- ще бъдат определени ресурсите, необходими за изпълнение на дейностите в срок и с необходимото качество;
- ще бъде сформиран екип за изпълнение – експерти, които ще извършат дейностите по внедряване;
- ще бъде извършен преглед на изготвената при планиране на тестването Спецификация на тестовите за приемане на системата, включително и на изготвената процедура за връщане на системата в последно работоспособно състояние (Rollback Procedure) и ако е необходимо, ще бъде актуализирани обхвата и вида на планираните тестовите за приемане;
- ще бъде разработен образец на протокол от проведени тестове за приемане на системата, в който ще се отразяват резултатите от тестовите;
- ще бъдат разработена процедурата за създаване на продукционната среда с подробно описание на последователността от действия;
- ще бъде подготвен образец на протокол за приемане на дейностите по изграждане на продукционната среда;
- ще бъде подготвен образец на протокол за внедряване за отчитане изпълнението на дейностите по внедряване;
- ще бъде изготвен детайлен график за изпълнение на дейностите по:
 - изграждане на продукционната среда;
 - внедряване и провеждане на тестове за приемане, съпроводени с представяне пред екип на Възложителя на реализираните функционалности с конкретна информация за период на изпълнение и експертите, които ще извършат дейностите.

Всички дейности във фазата на планирането ще бъдат съгласувани с Възложителя, като част от тях ще бъдат изпълнени със съдействието на екипа на Възложителя.

Резултатите от дейностите по планиране внедряването на системата ще бъдат обобщени в План за внедряване, който ще бъде представен на Възложителя за одобрение, в

2 екземпляра на хартия и в редактируема електронна форма (‘.doc’ или ‘.docx’ файлов формат). Планът за внедряване като минимум ще включва:

- Подготовка за внедряване – описание на подготвителните дейности, които трябва да бъдат извършени преди внедряване;
- Процедура за внедряване – детайлно описание на процедурата и дейностите, които ще бъдат извършени, с подробни указания за изпълнение на всяка стъпка;
- Процедура за изграждане на продукционна среда с приложено ръководство за администратора за инсталация и конфигуриране на системата с подробни указания от типа „стъпка по стъпка“ за последователността от действия за подготовка на продукционната среда
- Екип за изпълнение – списък на експерти с данни за контакт (телефон, мобилен номер, е-мейл адрес, адрес за кореспонденция и т. н.);
- Спецификация на тестове за приемане на внедряването, включително и на изготвената процедура за връщане на системата в последно работоспособно състояние (Rollback Procedure);
- Образец на протокол от проведени тестове за приемане на внедряването, в които ще се отразяват резултатите от тестовете;
- Образец на протокол за внедряване, който ще послужи за отчитане изпълнението на дейностите по внедряване;
- Процедура за създаване на продукционната среда с приложено ръководство;
- Образец на протокол за отчитане изпълнението на дейностите по изграждане на продукционната среда;
- Детайлен график за изпълнение на дейностите по внедряване.

✓ **Изграждане на продукционна среда**

Изграждането на продукционната среда, в която ще функционира Системата, ще се извърши в съответствие с одобрения План за внедряване и разработената Процедура за създаване на продукционна среда, която като минимум ще включва следните действия:

- инсталация и настройка на операционната система и друг системен софтуер;
- инсталация на компонентите (софтуерни компоненти и база данни);
- настройки на конфигурационните файлове;
- изпълнение на сервизни скриптове, ако е необходимо и т. н.

За изпълнение на дейностите по изграждане на продукционната среда ще бъде изготвен протокол.

✓ **Изпълнение на дейностите по внедряване**

Цялостното внедряване на компонентите ще бъде проведено по одобрените от Възложителя План за внедряване и детайлен график за изпълнение на дейностите.

Процедурата по внедряване на компонентите по предварителна оценка като минимум включва:

- параметризация на базата данни, включително създаване на системните класификатори, конфигуриране на системата;
- зареждане в базата данни на „плоски файлове“ с агрегирани данни;
- провеждане на тестове за приемане на внедряването и отразяване на резултатите в съответния документ.

Тестовите за приемане на внедряването ще бъдат извършени съвместно с екипа на Възложителя, като резултатите ще бъдат отразени в протокол.

✓ **Процедура за създаване на продукционната среда**

Описание ще включва подробна информация за процедурата и действията по създаване на продукционната среда. Към описанието ще бъде приложено ръководство за инсталация и конфигуриране за администратора с подробни указания стъпка по стъпка за действията, които трябва да бъдат извършени.

Действията по изграждане на продукционната среда, в която ще функционира системата като минимум ще включват:

- инсталация и настройка на операционната система;
- инсталация на система (софтуерни компоненти и база данни);
- параметризация на базата данни;
- настройки на конфигурационните файлове;
- изпълнение на скриптове, ако е необходимо и т. н.
- подпис на протокол за отчитане изпълнението на дейностите по изграждане на продукционната среда.

В хода на планиране на внедряването с Възложителя ще бъде обсъден подход за параметризация на системата.

Процедурата и дейностите за изграждане на продукционната среда ще бъдат специфицирани при планиране на внедряването, ще бъде разработено и ръководството с детайлна информация за действията, които следва да бъдат извършени с подробни указания от типа „стъпка по стъпка“. Процедурата и приложенията ще бъдат представени на Възложителя за одобрение като част от Плана за внедряване.

✓ **Образец на протокол за изграждане на продукционна среда**

Най-общо протоколът ще съдържа информация за:

- Възложител;
- Място на изпълнение на дейностите по изграждане на продукционна среда;

- Представител(и) на Възложителя, присъствал(и) и осъществил(и) контрол на изпълнение на дейностите по изграждане на продукционна среда;
- Експерт(и), извършил дейностите по изграждане на продукционна среда;
- Период на изпълнение;
- Извършени действия – по процедурата за изграждане на продукционна среда и допълнителни дейности, ако е възникнала необходимост от такива;

Образецът ще бъде разработен в окончателен вид при планиране на внедряването и ще бъде представен на Възложителя за одобрение като част от Плана за внедряване.

✓ **Образец на протокол за провеждане на тестове за приемане на системата**
Най-общо протоколът ще съдържа информация за:

- Възложител;
- Място на изпълнение на тестовете за приемане на системата;
- Представител(и) на Възложителя, участвал(и) в провеждането на тестовете за приемане на системата;
- Експерт(и), провели тестовете за приемане на системата;
- Таблица с планираните тестове, като се всеки тест ще има описание на критерият, въз основа на който се оценява резултатът от изпълнение. Например:

№ по ред	Тест	Критерий за успешно изпълнение	Получен резултат (да/не)
1	2	3	4
1.			
2.			
3.			
4.			

Забележки:

1. Колони 1-3 се попълват от Сمارт Системс при планиране на тестовете за приемане на системата.
2. Колоната 4 се попълва от представители на Възложителя при провеждане на тестването.

Образецът ще бъде разработен в окончателен вид при планиране на внедряването и ще бъде представен на Възложителя за одобрение като част от Плана за внедряване.

✓ **Образец на протокол за внедряване**
Най-общо протоколът ще съдържа информация за:

- Възложител;
- Място на изпълнение на дейностите по внедряване;

- Представител(и) на Възложителя, присъствал(и) и осъществил(и) контрол на изпълнение на дейностите по внедряване;
- Експерт(и), извършил дейностите по внедряване;
- Период на изпълнение;
- Извършени действия – по процедурата за внедряване и допълнителни, ако е установен необходимост от такива;

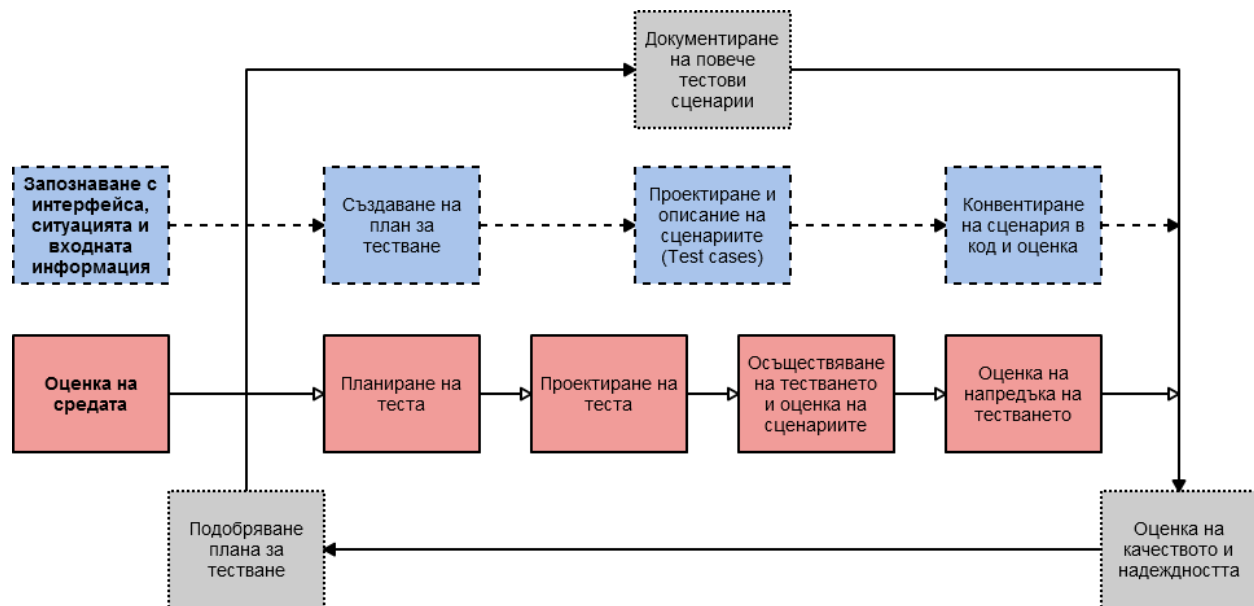
Образецът ще бъде разработен в окончателен вид при планиране на внедряването и ще бъде представен на Възложителя за одобрение като част от Плана за внедряване.

7.2.3. Методология за тестване

Смарт Системс ще проведе тестване на софтуерното решение в създадената тестова среда, с цел да потвърди, че разработеното решение отговаря на функционалните и нефункционалните изисквания. Това се постига чрез осъществяване на следните подцели на тестването:

- Откриване на всички грешки в кода, които екипът трябва отстранява;
- Откриване на грешки при дизайна;
- Откриване на повреди от неочаквано потребителско поведение;
- Тестване на всички елементи на решението.

Разработения софтуер ще се тества обстойно, за да се провери дали покрива изискванията на Възложителя. На следващата диаграма е показан процеса на тестване, който екипът на Смарт Системс ще следва:



Фигура 26 Процес на тестване

Процесът на тестване съпътства изпълнението на проекта и се състои от етапите: Планиране, Анализ и проектиране, Реализация и изпълнение, Анализ на резултатите и Заключение на дейности.

7.2.3.1. Тестов план

Настоящият тестов план е предварителен и ще бъде надграден във фазата на анализ.

Етапи на тестване:

1) Планиране

В етапа на Планиране се определят целите, използваните техники и методология за тестване, извършва се планиране и разпределение на ресурсите, изготвяне на график за провеждане на тестването, подготовка и приемане на тест план. Тест планът включва кратко описание на типовете тестване, базирани на анализа на изискванията, описание на различните тестови среди, структура на тест екипа, времевата рамка на тестовите задачи. Изборът на подходяща методология за тестване се базира главно на определяне на основните модули, под модули и компоненти на програмната система и идентифициране на критичните точки за бизнеса и отделните групи потребители на системата.

2) Анализ и проектиране

В етапа на Анализ и проектиране се определя последователността на тестовете и изискванията към тестовата среда, проектиране на тестовете и подготовка на тестови данни (валидни и невалидни).

Проектирането на тестове включва:

- Определяне на групите свойства (features) на програмната система;
- Определяне на основните части и под части на програмната система с цел по-лесно проектиране на тестовете чрез разделяне на множества, ориентирани към съставните части;
- Определяне на критичните точки за бизнес процесите, реализирани в програмната система;
- Определяне на типичните ежедневни сценарии за работа на различните групи потребители на програмната система;
- Дефиниране на критичните свойства (critical features), които трябва да бъдат тествани многократно през процеса на разработка;
- Дефиниране на задължителните свойства (required features), които трябва да бъдат тествани на отделни фази през процеса на разработка;
- Дефиниране на допълнителните свойства, подпомагащи процесите в програмната система (additional features), които могат да бъдат тествани в зависимост от времето и ресурсите;
- Определянето на критериите за приемане на програмната система (acceptance criteria).

След изготвянето им те се обсъждат с Възложителя. На базата на договорените критерии за приемане на системата тест екипът подготвя План за провеждане на

приемателни тестове на системата. Този план бива съгласуван и одобрен от Възложителя и след това става база за проектиране на приемателните тестове (acceptance testing).

Подходът при проектиране и подготовка на тестови случаи е тясно обвързан с изискванията към системата и се изпълнява в следната последователност:

- Определяне на основните части и под части на програмната система – Така се постига по-гъвкаво и ефикасно проследяване на тестовете за функционалното покритие на свойствата на програмната система (т.н. functional coverage).
- Идентифициране на критичните точки за бизнеса и отделните групи потребители на програмната система – По този начин се определят критичните точки за бизнес процесите, реализирани в програмната система (т.н. business critical points), както и типичните ежедневни сценарии за работа на различните групи потребители на програмната система (т.н. everyday business scenario).
- Определяне на групите свойства на програмната система (critical, required additional features) – Това значително подпомага процеса на разработка и тестване.

Избор на подходящи техники за проектиране на тестовете – Тази дейност включва преценка за спецификата на програмната система: дали е публична или критична откъм сигурността; доколко е сложна, комплексна или обикновена; има ли специфика във входните тестови данни; кои от избраните техники за проектиране на тестовете тест екипът владее добре.

Всички изготвени тестови сценарии подлежат на одобрение от оторизиран представител на Възложителя като след това стават база за проектиране на тестовете.

3) Реализация и изпълнение

Смарт Системс ще подготви тестова среда на сървъри предоставени от Възложителя.

Етапът на Реализация включва избор на тестове, генериране на тестови данни, изготвяне на тестовете и реалното им изпълнение. Реализацията на тестването включва:

- Избор на тест;
- Уточняване на основен и алтернативен начин на изпълнение, както и типичните изключения;
- Създаване на тест процедури (валидни комбинации на тестовете с подходящи тестови данни);
- Изпълнение на одобрените тест процедури.

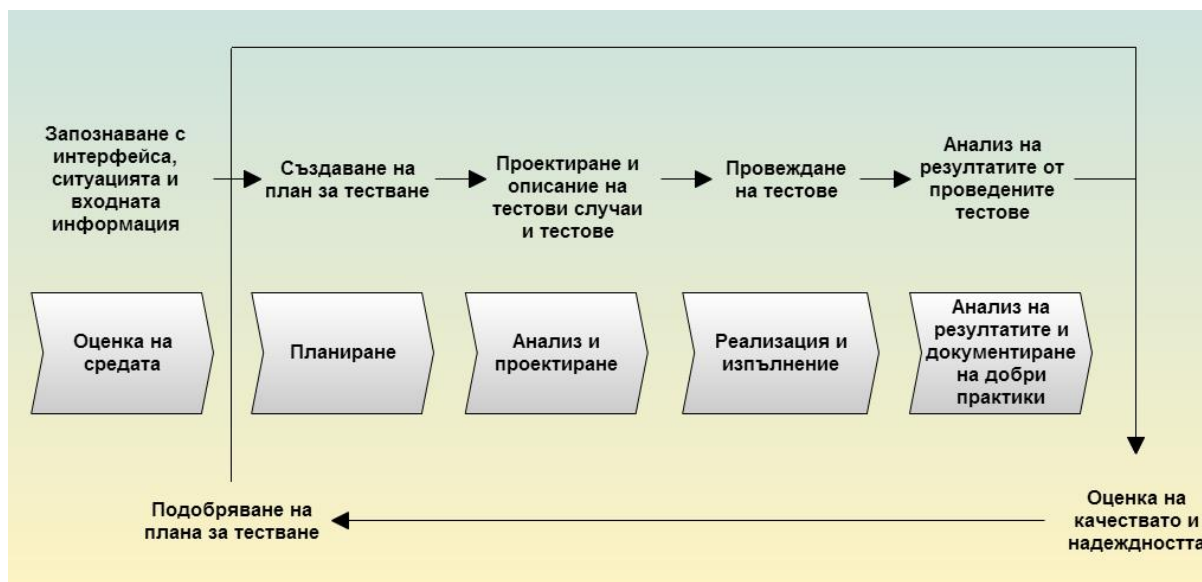
4) Анализ на резултатите

Етапът на Анализ на резултатите се състои от отчитане на получените резултати в избрания формат и проверка на условията за завършване на тестовете.

5) Заключителни дейности

Заключителните дейности обхващат изготвянето на обобщени справки, описание на добрите практики, оценка на проекта с цел подобряване на фирмените тестови процеси, архивиране на материалите.

Обобщените дейности по реализиране на тестовия процес са схематично представени на следната фигура:



Фигура 27 Обобщени дейности по реализиране на тестовия процес

Съществуват четири основни понятия, свързани с подготовката на тестовите случаи, реализацията на тестването и изграждането на тестовата среда:

Тестов случай (Test case)

Тестовият случай има за цел да тества поведението на определен модул (клас) в дадена ситуация. Обикновено се реализира чрез тестова функция. Всеки тест поставя тествания обект в подходящо за теста състояние, след което следи поведението и резултатите му в тестваната ситуация. Тестовите случаи извършват същинското тестване и трябва да са независими един от друг и от реда им на изпълнение;

Тестово множество (Test suite)

Тестовото множество се състои от набор от тестови случаи, които са логически свързани. Има за цел да тества поведението на определен модул (клас) в различни ситуации. Показва кои тестове са логически свързани. Реализира се чрез тестов клас, който има достъп до всички данни на тествания. Всеки тест поставя тествания обект в подходящо за теста състояние и следи поведението и резултатите му.

Тестова база (Test fixture)

Тестова база представлява обвивка на тестови множества. Има за цел да създаде необходимите условия за провеждане на тестовете. Отговорен е за инициализиращите и завършващите действия, както и за същинското изпълнение на тестовете. Отново, тестовете трябва да са независими един от друг и от реда им на изпълнение. Реализира се чрез йерархия от тестови класове. В базовите класове се дефинират общите операции, свързани с тестването (инициализация и завършване на теста). Те се предефинират в наследниците (тестови множества), ако е необходимо. Всеки тест поставя тествания обект в подходящо за теста състояние, инициализира го по подходящ начин, провежда теста и разрушава

тествания обект. Изключително важно е, тестваната система (компонент) да възстанови първоначалното си състояние след завършване на теста.

Тестова среда (Test harness)

Тестовата среда съдържа тестовата база и предоставя условия за създаване, добавяне и изтриване на тестови множества, както и за провеждане на съответните им тестове. Разполага със средства за контрол на тестването, запазване и анализ на резултатите.

Като добри практики при подготовката на тестови случаи може да посочим:

- Идентифициране на основните функционалности и идентифициране на ключовите функционалности за бизнес процеса;
- Идентифициране на основните части и под части на програмната система с цел по-лесно проектиране на тестове чрез разделяне на множества, ориентирани към отделните модули;
- Идентифициране на основните групи тестове и тестови сценарии за итерация със системата;
- Идентифициране на критичните функционалности (critical features), които трябва да бъдат тествани многократно през процеса на разработка;
- Идентифициране на задължителните функционалности (required features), които трябва да бъдат тествани на отделни фази през процеса на разработка;
- Идентифициране на допълнителни функционалности, подпомагащи процесите в програмната система (additional features), които могат да бъдат тествани в зависимост от времето и ресурсите;
- Документиране на тестовите сценарии;
- Използване на различни графични инструменти за описание на тестовите сценарии и инструменти за създаване на обвързаност между различните тестови сценарии и обвързаност между тестови сценарии и системна функционалност;
- Подготовка на тестови данни (валидни и невалидни);
- Определяне на условията за провеждане на тестовите сценарии;
- Определяне на последователността на тестове и изискванията към тестовата среда, проектиране на тестове и подготовка на тестови данни
- Определяне на критериите (изискванията към резултата от проведения тест) за преминаване/не преминаване на всеки тест.

Видове тестове залегнали в тестовия план

- Единица тестване – изпълняван се за най-малките тестови софтуерни единици – класове и методи;
- Компонентно тестване – изпълнява се за индивидуални компоненти и модули за да увери, че те коректно реализират бизнес функционалността;

- Системно тестване – след като компонентите и модулите са обединени системата се тества като цяло:
 - Функционално тестване;
 - Тестване на потребителския интерфейс;
 - Тестване на интерфейсите с останалите системи;
 - Тестване на производителността;
 - Тестване на сигурността и контрола на достъпа;
 - Тестване на възстановяемостта на системата след срив;
 - Тестване конфигурацията;
 - Регресивно тестване.

Цели на тестването

- Цели на Единица (Unit) тестването - проверка на единицата (клас или метод) според Дизайн модела и Модела на Имплементацията. Проверка на правилната обработка на въведените данни и получаване на очакваните резултати от всяка единица. Тестват се класовете с ключова функционалност;
- Цели на компонентното тестване - целите на компонентното тестване са изпълними прототипи, като резултата на всяка итерация. Функционалността на компонентите се тества и за тяхното съответствие с бизнес изискванията и стандартите;
- Цели на системното тестване.
 - Функционално тестване - основна цел на Функционалното тестване е Модела на потребителските случаи. Разработват се тестови случаи на основата на този модел като стремежът е да се покрият голямо количество случаи с различни комбинации от входно- изходни данни. Тези тестови случаи могат да служат и като метрики за напредъка на проекта;
 - Тестване на потребителския интерфейс - проверка за наличието на всички полета от формите и проверка за дължината и типовете на данните в тези полета; проверка, дали интерфейса съдържа всички списъци от предефинирани стойности, специфицирани в документацията; проверка на правилната навигация между екраните; проверка за поведението на интерфейса при въвеждане на невалидни и валидни данни; проверка на изгледа на интерфейса;
 - Тестване на производителността - проверка времето за отговор; проверка на времето на отговор за оперативните справки; проверка на времето на разпространение на данните;
 - Сигурност и тестване контрола на достъп - защита на данните при трансфера им до АМ; защита на данните при разпространението им; защита от

неоторизиран достъп да системната функционалност; валидация на различните нива на достъпа на данни; валидиране на журналите;

- Тестване при възстановяване след срив - тестване на времето за възстановяване след срив; проверка на необработените съобщения по време на срив;
- Тестване на комуникационната инфраструктура между модулите и връзката с външните потребители;
- Регресионно тестване - тестване коректното функциониране на системата след промени, когато е създадена нова версия и нови характеристики са прибавени или са поправени съществени дефекти.

Описание

- Единица тестване - най-ниските нива на единиците се тестват първи и тогава се използват за тестване на по-високите нива. Операцията се повтаря, докато не се стигне до най-горните нива.

За тестване на всяка единица:

Входен критерий	Веднага след имплементацията на ключова единица, тя трябва да бъде тествана от разработчика
Цел на теста	Да се осигури правилна навигация, вход на данните, обработката им и получаване на очакваните резултати
Техника	Изпълнение на всяка функция, използвайки валидни и невалидни данни. Необходимите съобщения за грешки се генерират, при вход на невалидни данни.
Необходими инструменти	Инструмент за автоматизирано тестване Инструмент за проверка на кода Инструмент за следене на грешките
Критерии за завършване	Всички планирани тестове са изпълнени. Откритите дефекти са коригирани.

- Компонентно тестване - тестването на компонентите е подобно на системното функционално тестване, свързано е с реализацията на потребителските случаи;
 - Системно тестване
- ✓ Функционално тестване - базирано е на метода на „черната кутия”, въвеждат се данни и се следи изхода:

Входен критерий	След приключване на Модела на потребителските случаи, дефиниране на потребителския графичен интерфейс. Изпълним прототип съществува.
Цел на теста	<p>Да се осигури правилна навигация, вход на данните, обработката им и получаване на очакваните резултати.</p> <p>Да се увери, че цялата функционалност на потребителските случаи е представена и стандартите са спазени.</p>
Техника	<p>Изпълнение на сценарий на потребителски случай с валидни и невалидни данни за да се увери, че:</p> <ul style="list-style-type: none"> – Очакваните резултати се получават, при вход от валидни данни; – Необходимата грешка или съобщение се визуализират при невалиден вход; – Всяко бизнес правило е правилно приложено; – Всички стандарти са спазени.
Необходими инструменти	<p>Автоматизиран инструмент за тестови скриптове;</p> <p>Инструменти за генериране на данни;</p> <p>Средство за следене на дефектите.</p>
Критерии за завършване	<p>Всички ключови потребителски случаи са тествани;</p> <p>Всички изисквания на приложените стандарти са спазени;</p> <p>Всички критични дефекти са отстранени.</p>

✓ Тестване на потребителския интерфейс:

Входен критерий	Изискванията за потребителския интерфейс са дефинирани и съществува прототип
Цел на теста	<p>Проверка чрез навигация по приложението дали правилно отразява бизнес функциите и изискванията, включително от прозорец към прозорец, от поле към поле.</p> <p>Проверка, че обектите и характеристиките на прозорците, такива като менюта, размери, статус, фокус отговарят на стандартите.</p>
Техника	<p>Създаване на тестове за всеки прозорец за проверка на неговите обекти.</p> <p>Създаване на тестове за проверка поведението на екраните при вход на валидни и невалидни данни.</p>

Необходими инструменти	Автоматизиран инструмент за тестови скриптове; Инструменти за генериране на данни; Средство за следене на дефектите.
Критерии за завършване	Всеки екран отговоря на специфицираните софтуерни стандарти.

- ✓ Тестване на производителността - целта му е да провери изискванията за производителността до каква степен са удовлетворени. Изпълнява се многократно като се променя нивото на натовареност на системата. Изпълняват се след системните тестове във времето, на преди тестовете за конфигурацията и тестовете при срив на системата:

Цел на теста	Валидиране времето за отговор на системата при нормални условия.
Техника	Използват се наборите от тестови случаи разработени при функционалното тестване. Тестовете вървят на една машина и се повтарят за много клиенти.
Необходими инструменти	Автоматизиран инструмент за тестови скриптове; Инструменти за генериране на данни; Средство за следене на дефектите.
Критерии за завършване	Тестовете са изпълняват без грешки и времето за отговор е в рамките на границите зададени в допълнителните изисквания.

- ✓ Стрес тестове:

Цел на теста	Валидиране времето за отговор на системата при условие, че максималния брой потребители извършват действия със системата. Валидиране времето за отговор на системата при условие, че много потребители модифицират едни и същи данни.
Техника	Използват се наборите от тестови случаи разработени при функционалното тестване. Тестовете вървят на една машина и се повтарят за максимален брой клиент.
Необходими инструменти	Автоматизиран инструмент за тестови скриптове; Инструменти за генериране на данни; Средство за следене на дефектите.

Критерии за завършване	Тестовите са изпълняват без грешки и времето за отговор е в рамките на границите зададени в допълнителните изисквания.
-------------------------------	--

- ✓ Сигурност и тестване на контрола за достъп - фокусира се към две основни области на сигурността; сигурност на ниво приложение, включваща рестриктивен достъп до бизнес функциите; сигурност на ниво система, включваща проверка на потребителите:

Входен критерий	Стартира при наличието на прототип с вграден контрол на достъпа
Цел на теста	Сигурност на приложението: Проверка на операциите – Създаване, Четене, Редактиране, Изтриване в зависимост от правата на потребителите и техните роли. Системна сигурност: Достъп само за автентикирани потребители.
Техника	Идентифициране на всеки тип потребител и определяне да видовете разрешени функции. Създаване на тестове за проверка. Промяна на типа на потребителя и повторно изпълнение на тестовите.
Необходими инструменти	Автоматизиран инструмент за тестови скриптове. Инструменти за генериране на данни. Средство за следене на дефектите.
Критерии за завършване	Функционалните тестове преминават без грешка, а при неавтентикиран или неоторизиран опит за достъп се извежда съответното съобщение.

- ✓ Тестване на системата след срив и за възможности за възстановяване:

Входен критерий	След успешно преминаване на системните тестове.
Цел на теста	Проверка процесите на възстановяване – автоматизирани или ръчни и състояние на системата след срив и последващо възстановяване.

Техника	Използва се набора то тестови случаи за функционално тестване, като се пускат след: Спиране на тока на клиентската станция; Спиране на тока на сървъра на приложението; Спиране на тока на сървъра на базата данни; Прекратяване на мрежовата връзка.
Необходими инструменти	Същите инструменти като при Функционално тестване.
Критерии за завършване	След задействане на процедурата по възстановяване системата трябва да е в правилното състояние.

✓ Тестване на конфигурацията:

Входен критерий	След успешно преминаване на системните тестове.
Цел на теста	Валидиране и проверка , че функциите на приложението се изпълняват правилно в средата на конфигурацията.
Техника	Използване на системни скриптове. Отваряне, затваряне на различни приложения по време на тестовите или преди започването им.
Необходими инструменти	Същите инструменти като при Функционално тестване.
Критерии за завършване	За всяка конфигурация тестовите преминават успешно с необходимите резултати.

✓ Регресионно тестване - регресионното тестване не е отделен тип тестване. То е повторение на тестовите, с цел проверка коректната работа на системата след като в нея са въведени промени:

Входен критерий	Нова версия на системата след като в нея са направени промени.
Цел на теста	Проверка поведението на системата след внедряване на промените.
Техника	Използват се вече разработените тестови случаи, набори от случаи, данни и скриптове за повторно тестове на системата. Възможна промяна и на тестовите случаи или данни в зависимост от вида на направените промени.
Необходими инструменти	Същите инструменти като при Функционално тестване.

Критерии за завършване	за	Същите критерии като изпълняваните тестове.
-------------------------------	----	---

Контекст за извършване на тестовете

Източниците на информация за провеждане на тестовете:

- План за разработка на софтуерния продукт – включва очакваните дати за провеждане на тестовете, както и резултатите, които следва да се представят от тестването.
- Спецификация на изискванията – включва нефункционалните изисквания към системата.
- Визия – Във Визията са заложили основни нужди и характеристики и продукта
- Модел на дейностите - Моделът на дейностите включва Бизнес модел и Модел на потребителските случаи. Моделът на потребителските случаи представя предвижданите функции и среда на системата и отразява нейните функционални спецификации, а Бизнес моделът описва начина, по който се изпълняват бизнес потребителски случаи.
- Документ Софтуерна архитектура - представя комплексен архитектурен изглед на системата, използвайки за целта редица различни архитектурни разрези, показващи отделни нейни аспекти.
- Дизайн модел – обектен модел, който описва реализацията на потребителските случаи и служи за извеждане на Модела на имплементацията и неговия програмен код.
- Модел на данните – Моделът на данните е подмножество на имплементационния модел, което описва логическия и физически вид на постоянните (персистентни) данни в системата.
- Модел на имплементацията - Моделът на имплементацията събира на едно място компонентите и съдържащите ги имплементационни подсистеми. Компонентите включват както тези, които подлежат на предаване като отчетни резултати (например изпълнимите компоненти), така и тези, от които се извеждат предаваните компоненти (например файлове с програмен код).

7.2.4. Методика за изпълнение на дейностите по обучение

Внедряването на системата изисква да бъдат инструктирани ползвателите участващи в нейното практическо приложение, чрез провеждане на обучение.

Като минимум обучението включва представяне на целите, задачите, структурата и функционалностите на системата, начин за работа с нея, въвеждане на информация, видове доклади и справки, които тя може да генерира и т.н. Целта е да се създаде необходимия капацитет за работа със системата на всички въввлечени страни и за правилното функциониране на системата.

Смарт Системс 2010 ЕООД ще:

- изготви концепция на обучението, план за действие, учебна програма и съдържание на обучението, учебни материали в РРТ презентация/и и др. подходящи формати, да размножи комплекти с учебни материали за всеки участник;
- осигури обучители;
- осигури логистиката за цялостното организиране и провеждане на обученията.

За провеждането на обученията „Смарт Системс 2010“ ЕООД ще осигури за своя сметка:

- Необходимия хардуер;
- Необходимия софтуер;
- Зала за провеждане на обученията;
- Учебни материали;
- Лектори;

„Смарт Системс 2010“ ЕООД ще разработи график на обученията, ще съгласува датите и ще подготви обученията, като отчита възможностите и ангажиментите на Възложителя за участие на определените служители. За целта, „Смарт Системс 2010“ ЕООД ще отправи предложение до Възложителя, към което ще представи и план за обучение. Планът следва да бъдат одобрен от Възложителя.

Смарт Системс т ще разработи учебна програма и необходимите учебни материали, които ще размножи и предостави на обучаемите.

В обучението ще бъдат включени лекции и семинарни занятия с балансирано съотношение на теорията спрямо практическите упражнения. При провеждането на учебните занятия ще бъдат използвани интерактивни методи и техники на преподаване – ролеви игри, решаване на казуси, дискусии. Обучението ще включва практически упражнения, за провеждането на които ще бъде използване тестовата среда на системата с примерна база данни. Практическите упражнения ще се провеждат по тестовите сценарии, използвани във фазата на тестване. С цел придобиване на уменията за работа със системата ще бъдат подготвени и конкретни казуси, подобни на описаните в тестовите случаи, но без указания относно начина на изпълнение. От обучаемите лица ще се изисква да изпълнят необходимите действия, като използват най-подходящата според тях функция на системата. Изпълнението на поставените задачи ще бъде проверявано от лектора и ще послужи за оценка на усвоените знания. По този начин обучението ще имитира реални ситуации от ежедневието на обучаемите лица.

Обучението ще се осъществи на български език.

Примерна програма за обучение

Продължителност на обучението – 2 дни, по 8 учебни часа на ден;

Организационна форма на обучение – дневна присъствена форма

Примерният учебен план включва общи модули за представяне на проекта и на разработена ИСВТ, в които участват всички обучаеми лица, включително системните администратори, които ще поддържат системата.

Таблица 11 – Примерен учебен план

№	Тема	Продължителност (учебни часове)		
		Общо	лекции	упражнения
I.	Представяне на проект „ Развитие на информационна система „Външна търговия“ с цел добавяне на нови функционалности и миграция на платформата“	1	1	
II.	Надградена Информационна система „Външна търговия“– общо представяне	1	1	
IIIa.	Функционален обхват на ИСВТ – общо представяне	3	2	1
IIIб.	Функционален обхват на ИСВТ – обособени панели за специализирано обучение за потребители и администратори	10	6	4
IV.	Гаранционна поддръжка. Процедура за управление на възникнали проблеми.	1	1	
Обща продължителност		24	14	10

Примерна учебна програма

Таблица 12 – Примерна учебна програма

Тема	Тема	Продължителност (учебни часове)	
		Лекции	упражнения
I. Представяне на проект „ Развитие на информационна система „Външна търговия“ с цел добавяне на нови функционалности и миграция на платформата“	Цели. Дейности и етапи на изпълнение. Изисквания. Постигнати резултати	1	
II. Надградена Информационна система „Външна търговия“– общо представяне	Системна архитектура. Технологична платформа. Достъпност. Сигурност. Потребителски интерфейс. Интеграции с външни системи.	1	

Ша. Функционален обхват на системата – общо представяне	Въвеждане на данни за различните роли в системата. Обработка на данни. Корекция и анулиране на операции. Механизми за сигурност. Системни справки	4	2
Шб. Функционален обхват на ИСВТ – обособен панел за специализирано обучение на ИТ експерти и системни администратори,	Администриране на потребители. Регистриране на потребители и групи. Задаване на права за достъп. Видове настройки, допустими стойности. Управление на статистически класификации. Дефиниране и редактиране на менюта. Управление на образци за изследвания. Следене и регистриране на извършените промени Журнал на системата – съхраняване на информация за всички извършени промени в данните за отделните информационни обекти на системата. Генериране на справки за извършените промени за избран период и експортиране във външен файл.	2	2
Шб. Работа с база данни на ИСВТ – обособен панел за обучение на ИТ експерти и администратори	Администриране конфигурацията на системата, в това число управление на резервните копия на базите данни и архивирането; управление на системните логове и журнали; управление на уведомяванията, нотификациите, алармите и др. Архивиране и възстановяване на данни. Процедура за архивиране и възстановяване на данни.	2	2

IV. Гаранционна поддръжка.	Процедура за управление на възникнали проблеми.	1	
-----------------------------------	---	---	--

Учебни материали

Учебните материали ще са на български език и ще съдържат материали по предварително разработена програма за обучение.

Предоставените на хартиен носител и/или в електронна форма учебни материали ще отговарят на изискванията за публичност и информация.

Присъствен списък

По време на учебните занимания на обучаваните ще се предоставят предварително подготвените регистрационни форми, в които те ще могат да попълнят своите данни и ежедневно да отбелязват присъствието си. При провежданото обучение ще се използва семинарна форма на обучение в комбинация с практически занятия.

Присъствен списък
на участниците в
Обучител:.....

№	Име, Фамилия	Дирекция	Длъжност	e-mail	Период на обучението	Получени материали /подпис/
1.						
2.						
3.						

4.						
5.						
6.						
7.						
8.						
9.						
10.						

Анкетни карти

В края на обучението участниците ще попълват анкетни карти, които „Смарт Системс 2010“ ЕООД ще обобщи в доклад. Оценка на проведеното обучение може да се извърши, чрез анкетиране на обучаемите. На оценка ще подлежат: съдържанието и формата на обучението, придобитите знания и умения, системата на обучение, стила на преподаване, подхода на лектора, процеса и структурата на обучението, приложимостта на наученото в практиката. За целта се определят показатели, измерващи ефективността от проведените обучения (например степен на удовлетвореност на участниците в обучението, повишаване на качеството на работата на участниците след посещение на обучението, намаляване на допусканияте грешки, и др.) „Смарт Системс 2010“ ЕООД ще да приложи съвкупност от различни методи за оценка.

Имена:.....

Институция:.....

Позиция:.....

1. По мое мнение обучението беше:

- a) много успешно ☐
- b) добро ☐
- c) незадоволително ☐

2. По мое мнение продължителността на обучението беше:

- a) много дълга ☐
- b) много кратка ☐

с) нормална ☐

3. С оглед бъдеща дейност, свързана със системата, обучението беше:

а) ненужно ☐

б) необходимо ☐

с) много добро ☐

4. Според мен допълнително следва да се включи в програмата на обучението /моля да напишете Вашите предложения/:

а) Няма нищо, което да се допълни или промени в програмата ☐

б) (Не мога да преценя) ☐

с) (-----) ☐

5. По мое мнение раздадените писмени материали бяха:

а) на много високо ниво и достатъчни ☐

б) добре подготвени и достатъчни ☐

с) незадоволителни ☐

6. Общото ми мнение за организацията на обучението:

а) на много високо ниво ☐

б) задоволително ☐

с) незадоволително ☐

7. Коя част от обучението беше най-интересна и коя част – най-безинтересна?

Еднакво интересни са и двете части – първата обуславя прилагането на втората на практика
Темата като цяло беше интересна
Интересна – примери
Тестване на системата
Тестване на механизъм за оценяване на рисковия потенциал
Като цяло обучението беше много интересно

Всички

8. Моля да дадете Вашите идеи и препоръки във връзка с организиране на подобни обучения в бъдеще:

Не мога да преценя
Нямам препоръки
Много добра организация. Нямам препоръки
Нямам предложения
Нямам

Бележки:

ОЦЕНКА НА ОБУЧИТЕЛИТЕ

Име на обучителя	Презентация	Оценка			
		Незадоволителна	Добра	Много добра	Отлична
	Съдържание				
	Представяне				
	Практическа важност за Вашата работа				

7.3. Описание на архитектурата и подхода за реализация на предлаганото софтуерно решение

Работата по реализация на целта на проекта може да се обособи в две основни задачи:

- Миграция на съществуващата ИС „Външна търговия“ към използване на нова СУБД – Microsoft SQL Server, както и разработка на обновена версия на системата, притежаваща пълния обем функционалност наличен в текущата версия на системата, но базиран на гъвкава и лесно разширяема архитектура ориентирана към услугите (SOA), както и на модерна и водеща технологична платформа (.NET Framework)
- Надграждане на така изградената обновена версия на ИС „Външна търговия“ (ИСВТ), чрез добавяне на нови функционални възможности и разширяване на съществуващите такива

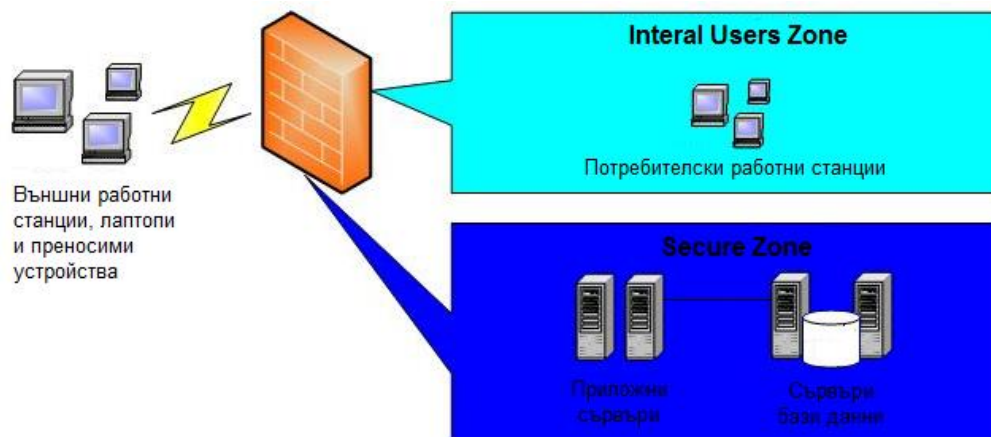
От архитектурна гледна точка, смятаме че ключово за успешната реализация на подобна модерна статистическа информационна система е не просто механично да се трансформира съществуващата функционалност в идентично приложение с нова платформа, а да се обособи от действащата функционалност ясно дефинирано и енкапсулирано модулно ядро от функционалност за управление на статистическа информация, на база на което да се базират конкретните компоненти на ИСВТ.

7.3.1.1. Архитектура

Обновената система ще бъде проектирана като централизирана уеб базирана система с архитектура ориентирана към услугите (SOA), комбинираща в себе си основните компоненти на текущата версия на ИСВТ, които от своя страна използват инфраструктура от стандартизирани услуги за реализация на обработката за конкретните типове статистическа информация.

Предлаганото разделение на реализацията на системата на ядро от стандартизирани услуги за обработка на статистическа информация и компоненти реализиращи конкретните функционални възможности на система, на база тези услуги цели да се постигнат изискванията за модулност, мащабируемост и гъвкавост, чрез която ще се реализират функционални модули, позволяващи модификация, допълване на нови модули или пълна подмяна на модули без необходимост от внасяне на изменения в останалите и в базисното ядро на системата.

7.3.1.1.1. Физическа архитектура на системата



Фигура 28 Физическа архитектура на системата

Предлаганото решение ще бъде адаптирано към съществуващата хардуерна, мрежова и приложна среда в НСИ, като в примерния вариант на физическа архитектура изложен в настоящото предложение, предлагаме да има като два основни физически компонента:

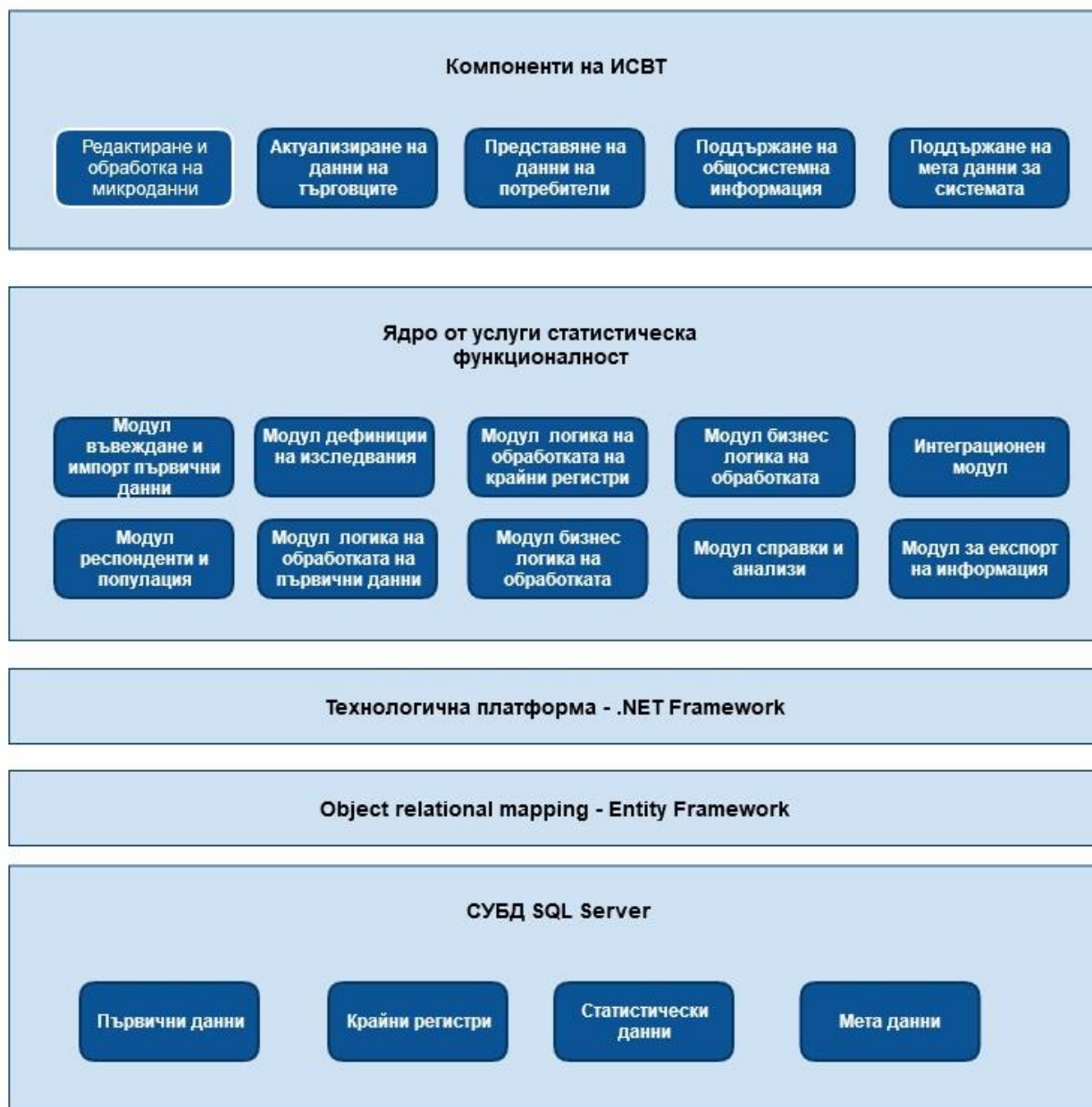
- Приложен сървър изпълняващ заявките към уеб приложението на системата, чрез който потребителите работят със системата
- Сървър за управление на базите данни на системата – с инсталиран описания в спецификацията като наличен Microsoft SQL Server Standard Edition.

Предлаганата физическа архитектура, ще бъде реализирана чрез създаване на виртуални сървъри чрез предоставения от Възложителя лиценз за платформата за виртуализация VMWare, на физическия хардуерен ресурс предоставен от Възложителя

Съображенията за максимално добро обособяване и осигуряване на информационна сигурност предопределят препоръчително отделяне на сървърите в т.нар. DMZ – зона от мрежата, която е изолирана от останалите части на вътрешната мрежа на НСИ отделена от външната среда с външна защитна стена, но и с вътрешна защитна стена, така че евентуален пробив да не доведе до риск за сървъра с приложението на системата или сървъра съхраняващ неговите бази данни.

Предлаганата физическа архитектура осигурява и възможност при решение на Възложителя реализацията да бъде използвана и в рамките на инфраструктурата на ДХЧО.

7.3.1.1.2. Предлагана логическа архитектура



Фигура 29 Логическа архитектура на предлаганото решение

Надградената ИСВТ ще представлява статистическа информационна система със следните основни възможности:

- Зареждане, коригиране, редактиране и обработване на първични данни;
- Поддържане на Първични и Крайни регистри на наблюдения - микроданни;
- Поддържане и актуализиране на данни за търговците;
- Поддържане и ползване на статистически класификации;
- Администриране на конфигурационни данни за логическия контрол на обработките;

- Поддържане на общосистемна информация, администриране на групи и потребители в релационната БД.

Реализацията на компонентите на надградената ИСВТ се базира на реализация на функционалност, която позволява управление на трите основни типа данни, които могат да бъдат класифицирани в системата:

- Микроданни – данни от единичните наблюдения, на които се базира статистическата обработка
- Макроданни – агрегирани данни, използвани при статистическите анализи
- Мета данни – данни описващи обработваната в системата статистическа информация.

Концепцията за реализация на системата е, че конкретните функционални компоненти за работа с микроданните няма да съдържат фиксирана (хард кодирана) логика за събиране, обработка и извличане на информация, а логика която определя поведението на реализацията на база на мета данните за съответния информационен обект.

7.3.1.1.3. Подход към реализацията на ядрото от услуги за обработка на статистическа информация

Ядрото от услуги за обработка на статистическа информация е свързано с обектно ориентирано проектиране на функционални възможности, обслужващи реализацията на основните стъпки на статистическия процес, а именно:

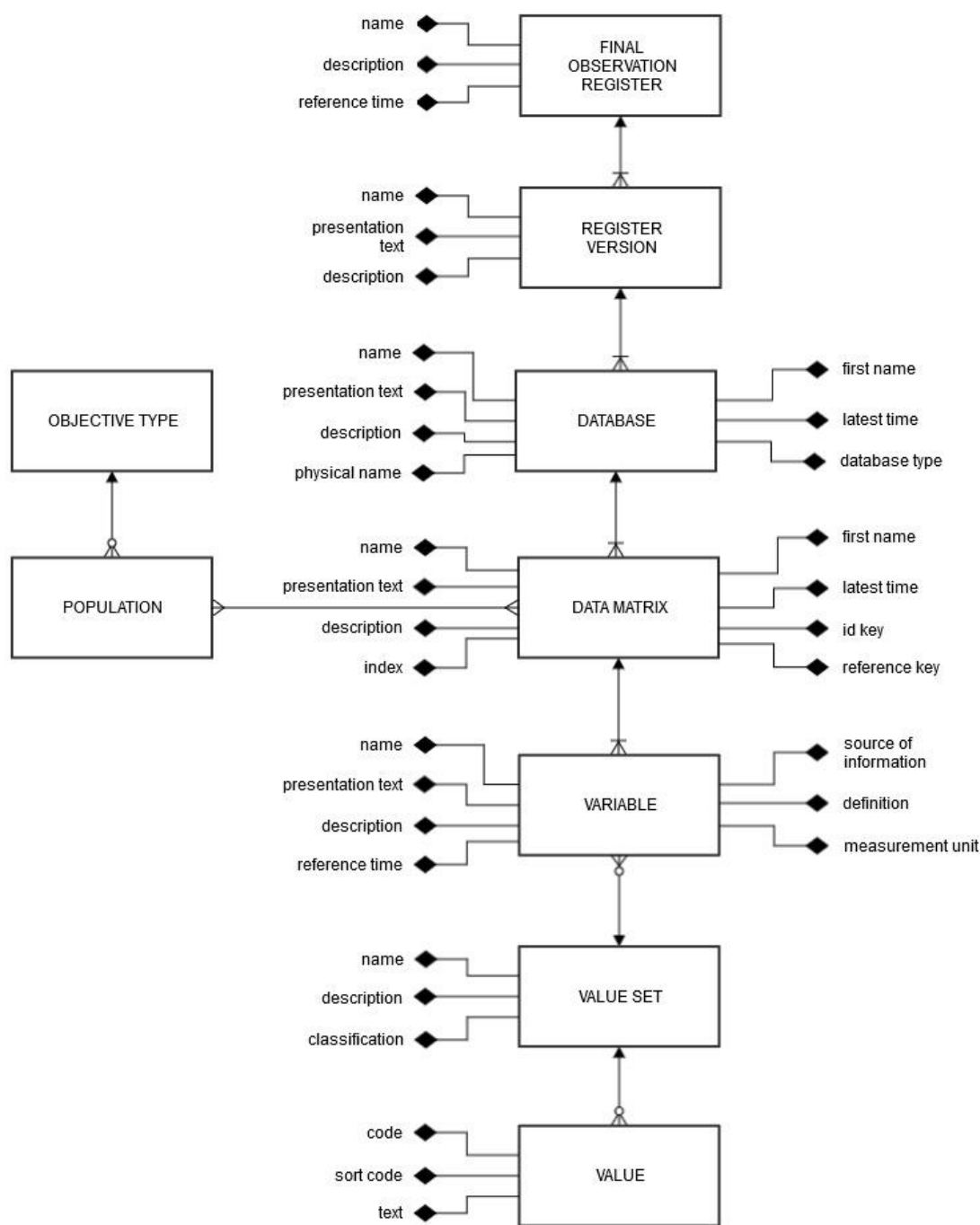
- Събиране на данни
- Съхраняване на данни
- Обработка на данни
- Разпространение на данни

Всеки един от елементите на процеса биват поддържани от различни функции в ядрото за статистическа обработка на системата:

- Събиране на данни – за да се поддържа нужната функционалност за реализация на процеса ядрото на системата ще реализира модули за управление на информацията за респонденти, популацията на която се базират изследванията, както и модул за дефиниране и управление на шаблоните за изследвания
- Съхраняване на данни – събраните първични данни биват съхранявани в оригинален вид, чрез поддържане на модул за регистър на първичните данни, а след обработката и изчистването им се съхраняват в модул за крайни регистри на наблюденията
- Обработка на данни – микроданните биват обработвани допълнително чрез модулите за логически контрол и функционалностите за агрегиране и допълнителна статистическа обработка на първични данни за реализация на модул „Структура на търговията“ и модул за „Вътрешен регистър на търговците“

- Разпространение на данни – поддържането на единен модул за представяне на данни за крайни потребители на статистическа информация, в който по унифициран начин се генерират експортните данни, следи се точния обем и съдържание на експортираната информация и се поддържат унифицирани средства за трансформирането на информацията от системата към различни формати, ще позволи ефективна поддръжка, добро качество на предоставяната информация и лесно преизползване на

Функционалните възможности за събиране на статистически данни в ядрото трябва да позволяват адаптивност на модела на събираните данни при постоянно променящите се изисквания и нужди на потребителите на системата. Една от основните причини налагащи използването на генерично ядро в системата е точно възможността за промяна в модела на събираната информация без регресионно отражение на останалите функции на системата и лесно надграждане на функционалността му. За целта реализацията на модела на данни на регистрите на първични данни ще се базира на общ концептуален модел на начина на структуриране на микроданните в системата:



Фигура 30 Концептуален модел за съхраняване на данни от изследвания

Подхода се базира на моделиране на всички получени резултати от изследвания, подлежащи на вписване в регистрите на наблюденията (observation registers), като общ клас съдържащ матрица на резултатите от наблюдението (observation matrix), съхраняващ информация за популацията (респондент, обект на наблюдението, извършил изследването)

в редовете на матрицата (MatrixRow) и стойностите регистрирани в наблюдението (MatrixColumn).

7.3.1.2. Подход за изграждане на надградената версия на системата

Уеб приложението на системата ще представлява пълно функционално уеб приложение, използващо подхода за реализация на уеб приложения MVC, предлагащо на потребителите единна точка за достъп до три основни групи функционалности на системата:

- Функционалността и модулите свързани със зареждането, валидацията, изчистването и обработка на данните (модул за импорт, модул за управление на данните)
- Модул за управление на мета данните
- Административен модул за общосистемна информация
- Функционалността за конфигуриране, изпълнение и представяне на резултата от реализираните в системата справки, отчети и dashboards – модул за справки и анализи

7.3.1.2.1. Подход при проектиране на архитектурата на уеб приложението

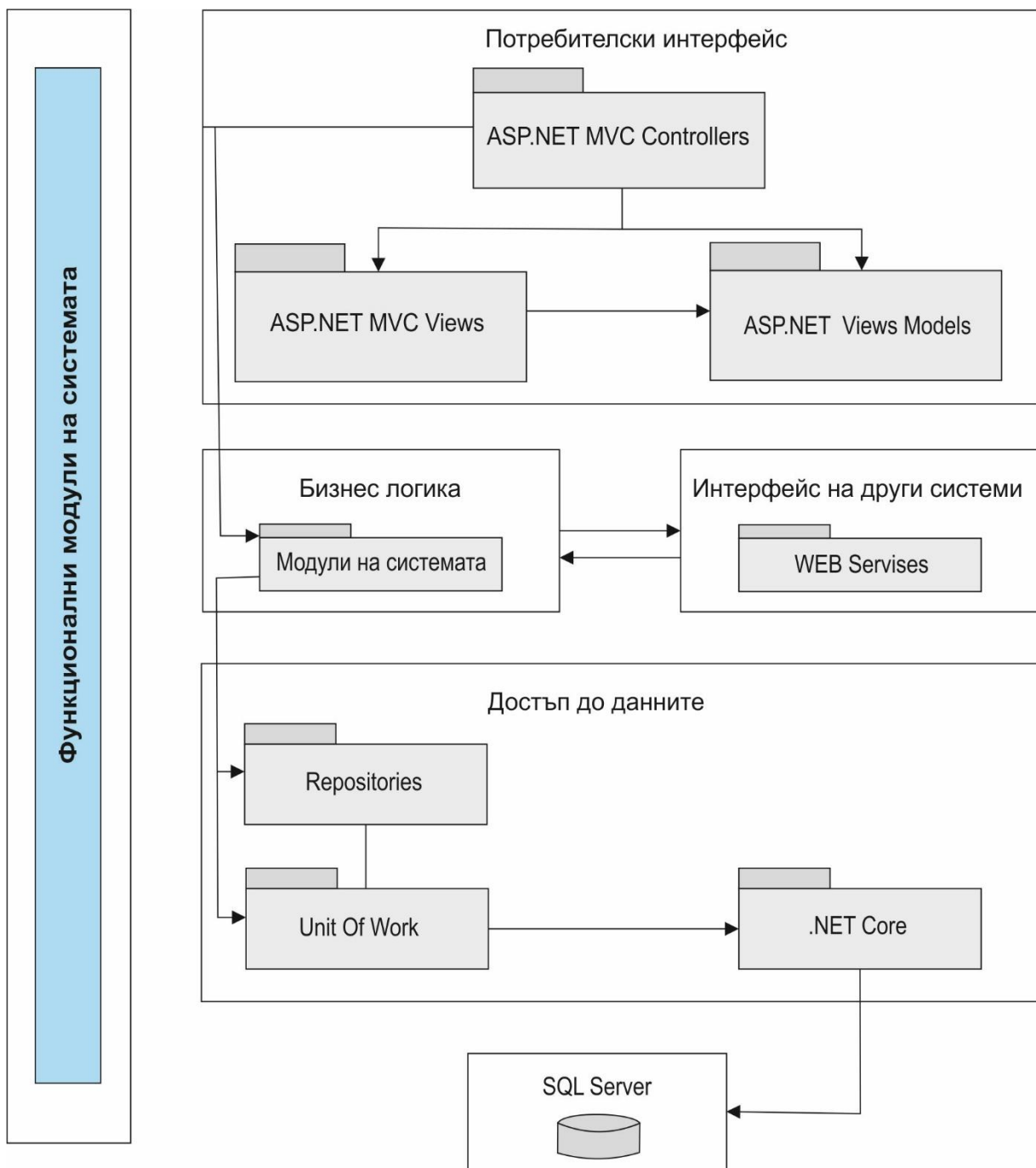
Отделните модули реализиращи функционалността на уеб приложението ще бъдат проектирани на база на най-добрите, съвременни и перспективни технологични платформи и архитектури, съгласно принципите на обектно-ориентирания анализ и дизайн. Обектно-ориентирания анализ и дизайн е софтуерен инженерен подход, който моделира системата като група от взаимодействащи си обекти. Всеки обект представя даден елемент от системата, която се моделира и характеризира със своето състояние и поведение. Проблемния домейн се декомпозира на отделни обекти следвайки принципите на дизайн от общото към частното и свързване на частите съобразно техните отговорности. Обектно-ориентирания анализ и дизайн по същество е процес на последователни действия на опростяване т.е. оперирайки с проблемния домейн в него се „инжектират” структури, които го декомпозират и опростяват. Структурите, които служат за декомпозиция и опростяване, представляват шаблони за дизайн, които са доказали своята ефективност при много различни ситуации и се препоръчват от водещите производители на софтуерни решения.

7.3.1.2.2. Слое (нива) на уеб приложението на системата

Предлаганото уеб приложение ще има архитектура която може да бъде декомпозирана на отделни слоеве (нива) комуникиращи помежду си по строго определени интерфейси. Основно предимство при този подход е, че позволява в отделните слоеве да бъдат извършвани значителни промени без това да оказва влияние на останалите, което води до изключителна гъвкавост. Слоевете са определени така, че да групират елементите, които искаме да можем да варираме независимо. Слоевете са определени така, че да групират елементите, които искаме да можем да варираме независимо. При централизираните системи доказан подход е разделянето на следните слоеве:

- Слой на базата данни;
- Слой на бизнес логиката;
- Слой на потребителския интерфейс (презентационен слой);

Всеки слой в последствие се декомпозира на отделни модули, като комуникацията между модулите се осъществява също по строго специфицирани интерфейси. Разделението на ясно разграничени слоеве и обособяването на слоя на базата данни от слоевете на бизнес логиката позволява да се осигури възможността цялостното решение да бъде съвместимо както със съществуващата инфраструктура в НСИ както и с виртуална инфраструктура, съответно върху Държавния хибриден частен облак.



Фигура 31 Системна архитектура на реализацията на ИС – трислойна архитектура MVC

Слой за достъп до базата данни

Задачата на слоя за достъп на базата данни е да обслужва и съхранява данните на информационната система. Уеб базираната система ще включва в себе си и система за управление на съдържанието, която ще бъде изградена при спазване на подхода за проектиране Repository design pattern, като функционалните механизми на системата няма да имат директен достъп за манипулиране на данните в базата данни, а ще използват

реализираните в слоя за управление на данните методи. Системата за управление на съдържанието от своя страна ще изпълнява подадените заявки за регистриране, промяна или извличане на информация БД, в съответствие с изискванията за разграничаване на достъпа, сигурност, както и осигуряване на консистентността на данните и неделимостта на транзакциите.

Слой за управление на данните ще бъде реализиран на базата на механизма за поддържане на обектно- релационни съответствия в Microsoft .Net Framework - Entity Framework. Използването на подхода за автоматизирано поддържане на съответствията между информационни обекти и релационната база данни осигурява няколко основни предимства:

Функционалността за съхраняване на данните става независимо от спецификата на съхраняващата база данни СУБД, поради факта че Entity Framework работи с различни популярни СУБД;

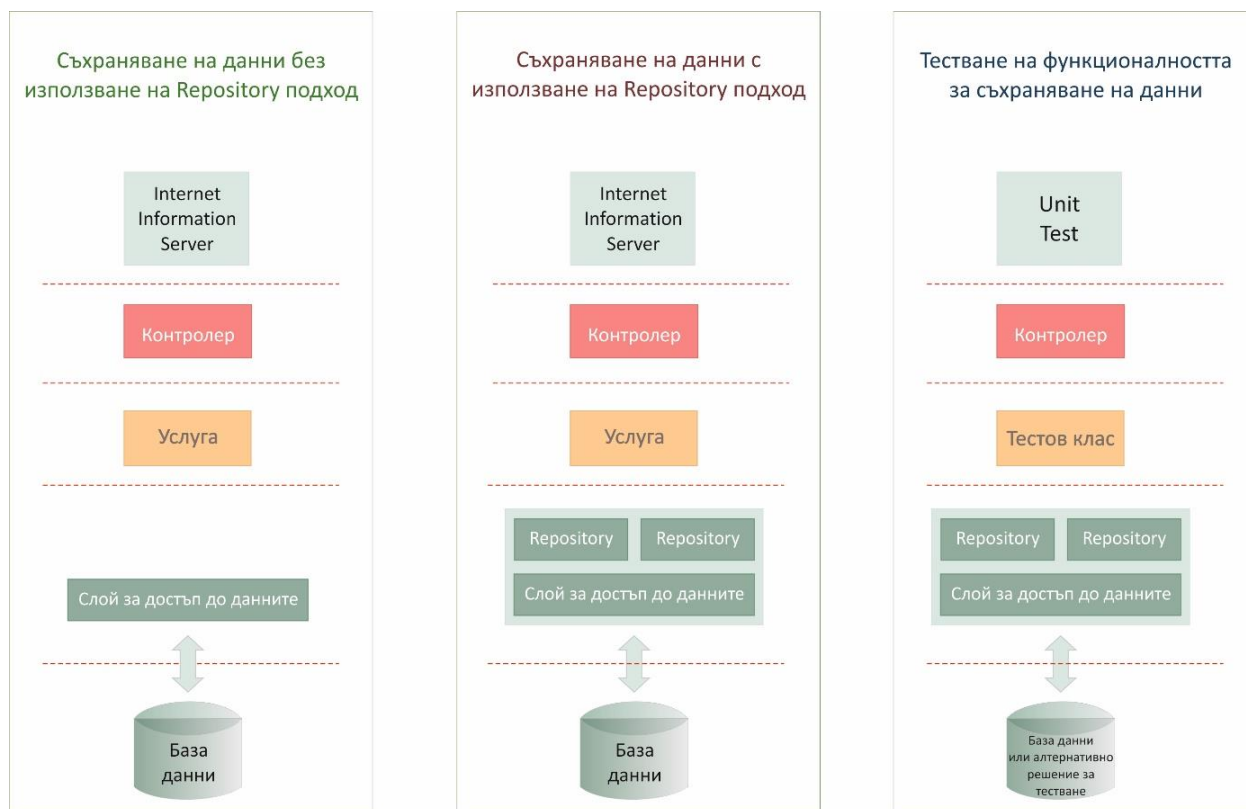
Промените в структурата на информационните обекти може да бъде автоматично отразена в релационния модел;

Допълнителния слой на абстракция позволява по голяма гъвкавост по отношение на физическия модел на данните – т.е. улеснява скалирането на базата данни и евентуалното му разпределяне в повече бази данни;

Entity Framework осигурява пълна поддръжка на транзакции на ниво на слоя за достъп до данните, така могат да се осигури изпълнението на заявката към базата данни на приложението, да отговорят на изискванията ;

- Atomicity – атомарност на транзакцията (при грешка се отменят всички заявки от транзакцията, за да е успешна транзакцията са изпълнени всички заявки от нея);
- Consistency – цялост на данните след всяка изпълнена транзакция;
- Isolation – изолация на данните по отношение на други транзакции;
- Durability – стабилност на транзакцията, липса на възможност за загуба на данни.

Основното предимство на избрания подход се изразява във възможността да не се влага логика в услугите използващи съдържанието на базата данни, тъй като слоя до данните му дава нужните методи и отделя със слой на абстракция по-ниските нива на работа с информационните обекти и базата данни. На следната схема са показани двете основни предимства пред традиционния подход, липсата на директно извикване на данните от бизнес логиката и възможността за автоматизирано модулно тестване на функционалността на приложението, дори преди да е завършена реализацията на базата данни:



Фигура 32 Съхраняване на данни

Слой за достъп до данните ще съхранява електронно съдържание в релационната база данни - Microsoft SQL Server. При проектирането и създаването на базата данни под внимание ще бъдат взети следните аспекти:

- Възможност за ефективна работа с големи обеми от данни;
- Възможност за осигуряване на ефективен механизъм за резервиране на данните;
- Високо ниво на сигурност.

Слой на бизнес логиката

Функцията на слоя на бизнес логиката е да обработва заявките за обработка получени от интерфейса на системата, да поддържа тяхната валидност (консистентност) спрямо идентифицираните бизнес процеси и правила и да осигури управлението на потока от данни, съобщения, и документи, свързани с работата на системата, като ги съхранява и извлича от слоя за достъп до данните. Според логическата архитектура на системата в слоя с бизнес логиката се разполагат всички модули, функции и процедури, които реализират функционалността на системата.

Реализацията на слоя на бизнес логиката ще включва набор от отделни компоненти които изграждат това ниво:

- От гледна точка на по-добра сигурност и гъвкавост на приложението вътрешната и външната част на приложението ще се обособят като отделни MVC приложения, които ще дават достъп до различна част от функционалността на системата, но ще се

реализират използвайки общата основа от модули и общ модел на данните, така че да се улесни максимално бъдещото развитие и на двете части на системата

- Бизнес логиката за въвеждане, актуализация и представяне на информацията от модулите на системата ще се реализира съгласно обектно ориентирания подход в гранулярно дефинирани независими компоненти – модули на системата, които си взаимодействат спрямо дефинирани между тях интерфейси. Модулите на системата са детайлно описани в секция „Функционални модули на информационната система“ на настоящото предложение. В допълнение ще бъде реализиран и модула за интеграция с външни системи, който ще позволява обмен на данни и изпълнение на части от функционалността на модулите на бизнес логиката при интеграция с дефинираните в техническото задание системи.

Слой на потребителския интерфейс

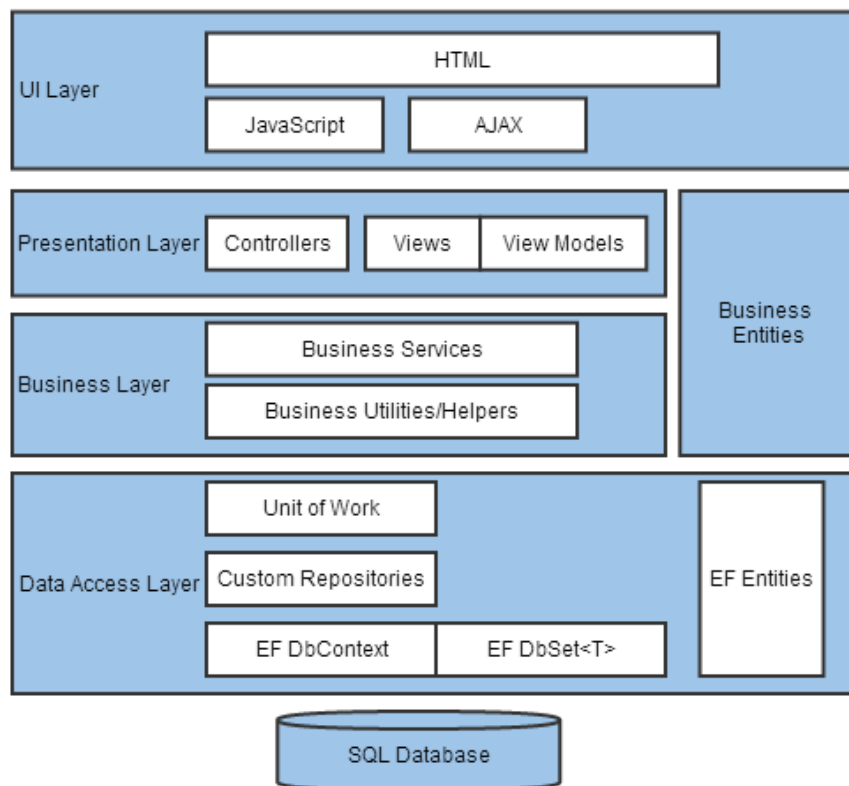
Задачата на слоя на потребителския интерфейс е да взаимодейства с потребителите на системата, така че да обслужва техните заявки и да представя необходимата им информация в определения формат. В този слой ще се организира и използва интерфейса на системата, от потребителите, за да извършват желаната от тях работа. Въведените данни в този слой се подават на слоя на бизнес логиката за последващата им обработка и съхранение в базата данни.

7.3.1.2.3. Прилагане на Model-View-Controller.

При реализацията на уеб приложенията на системата ще се използва архитектурния подход Model – View – Controller, чрез реализацията и в избраната технологична платформа .Net Framework, а именно ASP.NET MVC. При тази архитектура контролерите на приложението посрещат заявките на потребителите ги предават за обработка към бизнес логиката, попълват необходимите данни във View Model-ите и връщат съответното view, което предоставя визуализация на потребителския интерфейс. За реализирането на богат и удобен за работа потребителски интерфейс, който не изисква постоянно презареждане от сървъра view-то, което се рендира в брауъра ще бъде реализирано като се използва библиотеката за динамично уеб съдържание jQuery.

Слой

на
потребителския интерфейс ще отговаря на всички изисквания на техническото задание
към потребителския интерфейс.



Фигура 33 Многослойна Архитектура

Прилагането на MVC парадигмата осигурява следните преимущества:

Преизползваемост на компонентите. Разделението на модела от изгледа позволява компонентите на модела да бъдат преизползваеми за различни изгледи.

Скалируемост. Моделът осигурява както хоризонтална, така и вертикална скалируемост. Позволява промяна на броя потребители, транзакции, обръщения към системата. Функционалността на системата може да бъде променена, без промени във всички слоеве.

Гъвкавост. Моделът осигурява лесната заменяемост на отделните слоеве като поддържа капсулируемост на отделните нива и следи за еднопосочност на връзките между тях. Определя и входно – изходни точки между слоевете.

7.3.1.2.4. Предлаганата техническа платформа за изграждане на уеб приложението на системата – описание, обосновка и предимства

За реализация на софтуерните разработки в настоящата обществена поръчка Сمارт Системс ще използва платформата Microsoft .NET Framework. За език за програмиране ще бъде използван C#.

За среда за разработка ще бъде използвана средата Microsoft Visual Studio 2017.

Microsoft.NET Framework е популярна и съвременна платформа за реализация на бизнес приложения, която е специфицирана и създадена от Microsoft. Тя е базирана на отворени

стандарти, които са общодостъпни и са публикувани на страницата на ЕСМА. Възможностите на платформата заедно с наличните средства за разработка предоставят мощно средство за разработка, което позволява изграждане на съвременни архитектурни решения. .NET Framework е пуснат за разпространение през 2002г. От създаването си до момента платформата постоянно се развива и подобрява за да се утвърди като водеща технологична платформа за разработка на приложен софтуер.

Предложената платформа отговаря на следните изисквания:

- Бързодействие при изпълнение – програмният код не се интерпретира, платформата разполага със средства за компилация в оптимизиран за бързодействие платформено-зависим машинен код;
- Сигурност – програмният код е изолиран от хардуерната и софтуерната среда, в която работи, като по този начин го предпазва от програмни грешки;
- Лесна инсталация – инсталирането на софтуерните приложения не изисква сложни настройки;
- Поддръжка на утвърдени в индустрията езици за програмиране – MS.NET не е обвързана с конкретен програмен език;
- Отворени стандарти – MS.NET е базирана на отворени стандарти;
- Поддръжка на средства за централизирано съхранение на бизнес-обектите в релационна база данни;
- Поддържа уеб услуги (web services);
- Скалируемост – MS.NET притежава възможност за паралелна работа на множество сървъри, изпълнение на разпределени транзакции и управление на натоварването;
- Средства за мониторинг – т MS.NET притежава средства за наблюдение на натоварването, използваните ресурси, възникналите изключения и др.
- Среда за разработка - MS.NET е интегрирана с предложената среда за разработка MS Visual Studio.

4.1.1. Платформена независимост

MS.NET Framework, предоставя библиотека от класове FCL (Framework Class Library) и среда за изпълнение CLR (Common Language Runtime). Библиотеката от класове (FCL) осигурява множество от готови класове и интерфейси, които спомагат за ускоряване и оптимизиране на процесите по разработка и развитие на приложенията. За разработка под .NET Framework се използват езици от високо ниво, като създадения програмен код се компилира до платформено-независим междинен език, наречен CIL (Common Intermediate Language) код. По време на изпълнение на приложението CIL кодът (т. нар. „управляван код“) автоматично се компилира от CLR до изпълним код за конкретната хардуерна платформа и операционна система, на която работи приложението.

4.1.2. Бързодействие при изпълнение

Платформата разполага със среда за изпълнение CRE (Common Runtime Engine), която позволява приложенията да бъдат компилирани в междинен CIL (Common Intermediate Language) код. При изпълнението си този междинен код не се интерпретиран както при другите виртуални машини напр. Java, вместо това се извършва JIT (Just In Time) компилация в платформено-зависим машинен код (native code).

4.1.3. Лесна инсталация

Програмите, създадени на .NET Framework, както и техните компоненти, могат да бъдат инсталирани с просто копиране в желаната директория - процес, известен като XCopy Deployment.

4.1.4. Отворени стандарти

Спецификациите на CLI (Common Language Infrastructure), както и езиците C# и C++/CIL са стандартизирани от организацията ECMA и ISO и са отворени стандарти.

4.1.5. Поддръжка на утвърдени в индустрията езици за програмиране

Платформата Microsoft.NET не е обвързана с програмни езици. Тя осигурява езикова независимост. Това е възможно благодарение на съвместимостта на типовете данни, които отделните езици поддържат. CTS (Common Type System) дефинира всички базови типове данни, както и начинът, по който те могат да бъдат конвертирани един в друг. Тези типове са споделени между всички .NET езици и са стандартизирани в CIL. Към момента има създадени .NET версии на всички масово използвани езици за програмиране, в това число: C++, C#, Visual Basic и др.

4.1.6. Поддръжка на подсистеми за сигурност, базирани на утвърдени стандарти

Microsoft .NET Framework осигурява базови класове и среда за изпълнение (CLR), които са отговорни за поддържане на подсистемите за сигурност на платформата.

Платформата осигурява следните утвърдени стандартни подсистеми:

- Подсистема за управление на изпълнението на кода – Code Access Security;
- Подсистема за управление на достъпа на потребителите – Role Based Security.

Подсистемата за управление на изпълнението на кода определя нивото на сигурност, което е отредено да всяко парче код, което се изпълнява. Нивото на сигурност зависи от мястото на изпълнение, създателят на кода и от други фактори.

Подсистема за управление на достъпа на потребителите е базирана на потребителски роли, с помощта, на които за всеки потребител може да се определи дали има достъп до определен ресурс и дали ма право да извърши дадена операция. Управлението на потребителите с помощта на роли не зависи от начина по който .NET Framework идентифицира, автентикира и оторизира потребителите. Потребителите може да се автентикират и оторизират спрямо различни източници, в това число спрямо системата за сигурност на локален сървър, домейн или друг източник.

Платформата осигурява възможности за криптиране, генериране на криптографски ключове и хеширане на съобщения. Измежду поддържаните алгоритми са DES, SHA, AES, RC2 и други. Наред с криптографските възможности, платформата предоставя и средства за работа с цифрови сертификати.

Защитата на транспортно ниво може да бъде осигурена чрез осигуряване на защитени канали за комуникация през SSL или IPsec.

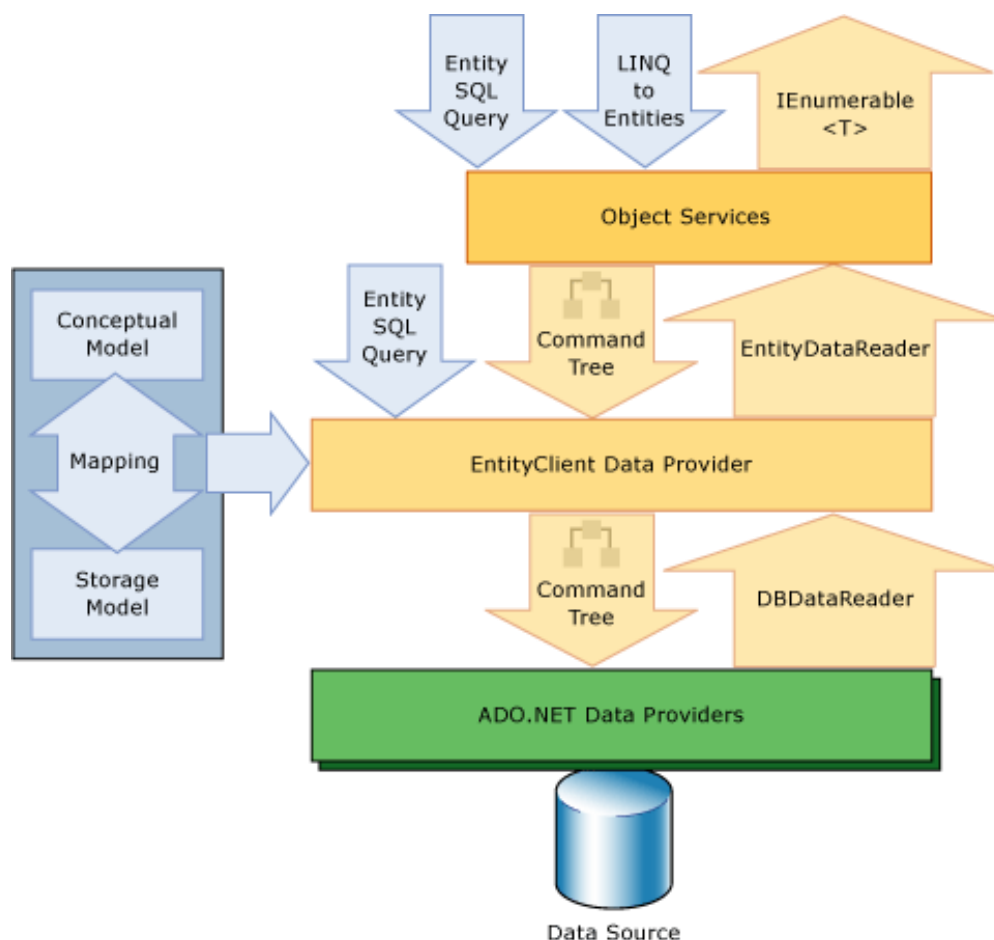
Сигурността на обменяните съобщения може да се гарантира чрез използване на WS-Security и чрез подписване и криптиране на всяко едно съобщение.

Програмният код, написан на .NET Framework се нарича управляван код, а също така и защитен код. Той се изпълнява в специална защитена среда наречена sandbox и е изолиран от хардуерната и софтуерната среда, в която работи. Това го предпазва от програмни грешки, които го правят уязвим за атаки, например препълването на буфера (buffer overflow). Платформата предлага система за сигурност, в която може да се интегрират всички използвани приложения.

4.1.7. Поддръжка на средства за централизирано съхранение на бизнес-обектите в релационна база данни

Microsoft .NET поддържа слените средства за централизирано управление и съхранение на обектите:

- ADO.NET Entity Framework – това е технология разработена от компанията Microsoft за достъп до данните. Технологията по същество представлява Object Relational Mapping (ORM), което позволява автоматизирана трансформация на релационните данни към проектирания обектен модел ползван от софтуерното приложение;
- LINQ (.NET Language Integrated Query) – това е технология интегрирана в езиците за програмиране, част от платформата .NET, унифицираща и улесняваща начина на достъп до информация, която по своята същност и дефиниция не е обектно ориентирана, напр. XML документ, релационна база данни и т.н. LINQ дава унифициран подход за изпълнение на заявки, който е приложим към разнообразни източници на информация, като резултатът автоматично се трансформира към ползвания от софтуерното приложение обектно-ориентиран модел.



Фигура 34 Централизирано съхранение и достъп до бизнес обектите

4.1.8. Поддръжка на утвърдени стандарти за обмен на данни с други системи

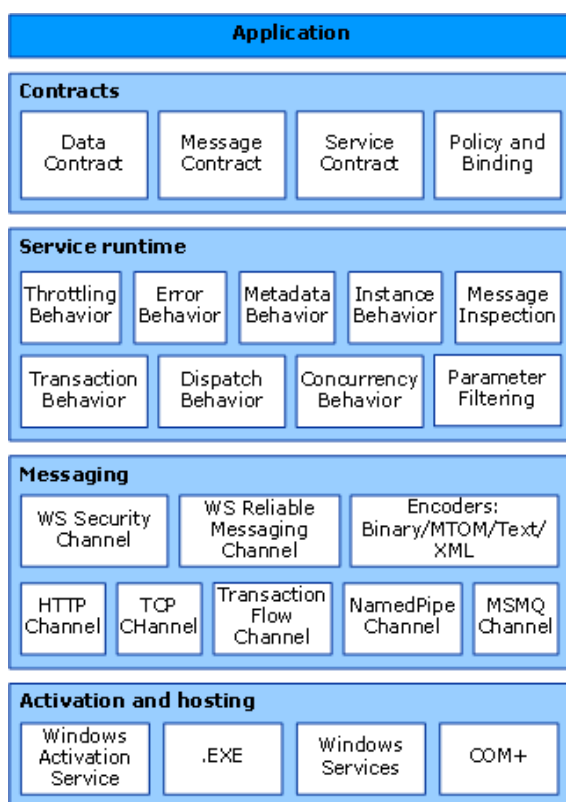
Обменът на данни с други системи ще бъде изграден с помощта на Windows Communication Foundation (WCF). WCF е надграждане на платформата (Microsoft .NET Framework), което е предназначено за изграждане на SOA решения със следните възможности:

- Широка съвместимост. WCF предоставя пълна поддръжка на всички протоколи, които са необходими за изграждане на инфраструктура от Web услуги (WS), в това число SOAP (Simple Object Access Protocol) за предаване на данни и изпълнение на конкретни операции;
- Интеграция с COM/COM+;
- Широк обхват на транзакцията. WCF поддържа следните начини за управления на транзакция: WS-AtomicTransactions, System.Transactions, MSDTC;
- Поддръжка на AJAX и REST;
- Различни начини за обмен на съобщения чрез създаване на контракти;

- WSDL(Web Service Description Language) за описание на интерфейса и предлаганите операции от една уеб услуги
- Сигурност. Технологията позволява използване на добре познатите стандарти SSL и WS-SecureConversation;
- Различни физически канали за обмен на съобщенията – HTTP, TCP, Named pipes и други.

4.1.9. Поддръжка на уеб услуги

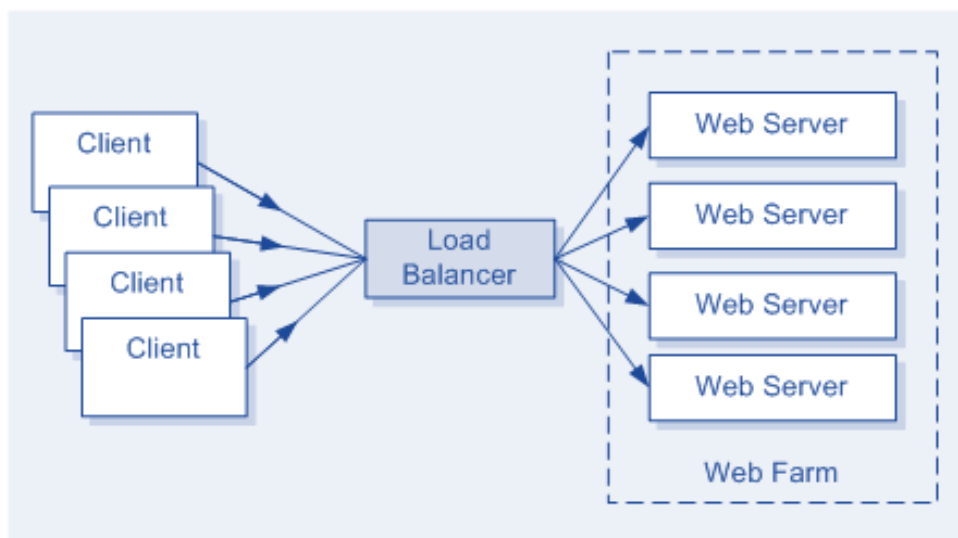
Част от Microsoft.NET е Windows Communication Foundation (WCF), която позволява да се създават интероперабилни уеб услуги. WCF покрива Web services (WS-*) стандартите, което позволява на приложенията да комуникират по платформено независим начин.



Архитектура на WCF

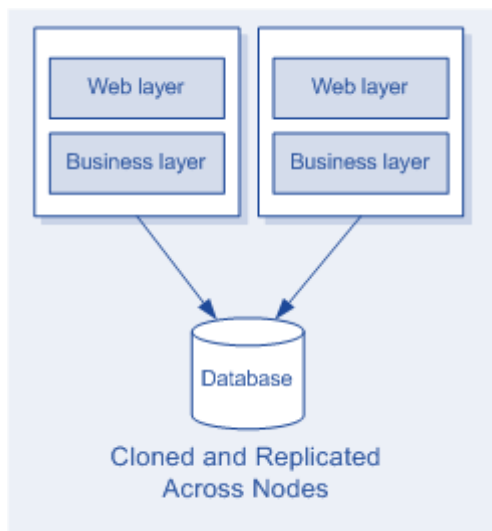
4.1.10. Скалируемост

Microsoft.NET разполага с вградени средства за осигуряване на скалируемост, възможност за паралелна работа на множество сървъри, разпределени транзакции и управление на натоварването. Microsoft.NET приложенията могат да работят във ферма от уеб сървъри в режим на разпределение на натоварването.



Управление на натоварването

Microsoft.NET дава възможност да се клонира даден приложен сървър, като по този начин софтуерната услуга може да се репликира в множество възли на дадена уеб ферма.



Разпределени транзакции

4.1.11. Средства за мониторинг

Microsoft.NET поддържа удобни средства и програмни интерфейси за мониторинг на ресурсите и приложенията. Това са:

- .NET Profiling API;
- Performance Counters;
- Performance Monitor;
- Windows Management Instrumentation (WMI).

4.1.12. Интеграция със средата за разработка Microsoft Visual Studio

Microsoft.NET е напълно интегрирана със средата за разработка Microsoft Visual Studio. Microsoft Visual Studio предоставя интегрирана среда състояща се от разнообразни инструменти и сървърна инфраструктура, която улеснява целия процес за проектиране и разработка на софтуерни приложения върху платформата .NET. Средата дава възможност за постигане на бързи бизнес резултати, чрез използването на продуктивен, предсказуем и адаптивен процес, предоставяйки прозрачност и проследимост през целия жизнен цикъл на разработка. Средата притежава мощни средства за проектиране, прототипиране, тестване и разработка, които позволяват бърз преход от създадената визия към реалното решение.

Microsoft Visual Studio води до увеличаване на производителността на екипа чрез използване на вградените модерни средства за сътрудничество, интегрираните инструменти за тестване и отстраняване на грешки, които позволяват лесното им откриване и бързото им коригиране, създавайки по този начин висококачествени решение с ниска производствена цена.

7.3.1.2.5. Предлагани средства (инструменти) за разработка на уеб приложението

За реализация на Системата ще се приложи обектно-ориентирания подход с използване на следните продукти и технологии в съответствие с най-добрите практики и разработки:

- ✓ Смарт Системс т ще реализира предлаганото решение на базата на система за управление на бази данни – Microsoft SQL Server, водещ продукт в областта на комерсиалните СУБД, сертифициран в съответствие с международния стандарт ISO/IEC 15408:2005, определящ т.нар. "Common Criteria for Information Technology Security Evaluation (CC)". Предлаганото СУБД Microsoft SQL Server отговаря напълно на техническата спецификация:
 - Поддържа сървър с повече от един процесор
 - Поддържа всички стандартни релационни типове данни, а също и собствени типове за съхранение на XML данни, текст, документи, изображения, аудио и видео данни, географски векторни и растерни данни;
 - Предлага надежден failover механизъм
 - Позволява инсталиране в клъстер
 - Предоставя графичен интерфейс за администрация на управляваните бази данни
 - Поддържа изгледи, които съдържат агрегирани стойности от една и повече таблици и да предоставя механизми за прозрачно обновяване на агрегираните стойности в момент на промяна на данните в изходните таблици;
 - Предлага ефективен начин за работа с големи обеми от данни, като позволява да се поддържа логическо разделяне на физическите таблици на няколко логически, с цел бързодействие;
 - Поддържа съдържание на кирилица
 - Пълно текстова индексация на съдържанието на файлове във формат Excel, Word, PDF, Text

- ✓ Ще се използва Visual Studio 2017, като средство за разработка, а C#.NET и JavaScript като езици за програмиране. Посочената развойна среда е снабдена с качествени средства за визуален дизайн, поддържа множество програмни езици и редактори на кода. Позволява разработка и отстраняване на грешки в многопотребителски сървърни приложения чрез обединената среда за разработка;
- ✓ SVN - като хранилище на кода, за поддържане на версиите на софтуера, документацията и друга информация по проекта Сمارт Системс използва внедрената при него система за управление на версия. Тази система позволява едновременна работа на членове на екипа върху едни и същи файлове, поддържа версии и под версии, позволява сравнение, възстановяване и обединяване на поддържаната информация.

Системата ще бъде разработена на базата на .NET Core 2.0 и .NET Standart 2.0 – платформи с отворен код, позволяващи на приложението да работи под различни операционни системи (MS Windows, OS X, Linux) с помощта на следните технологии:

- ✓ C# и JavaScript – езици за програмиране, използвани при разработката на приложението;
- ✓ Entity Framework Core ORM – технология за достъп до бази данни с отворен код;
- ✓ ASP.NET Core – технология за разработка на Уеб приложения с отворен код;
- ✓ XML, JSON – стандарт за обмен на данни;
- ✓ AJAX - група от свързани техники за разработка на Интернет приложения, позволяващи асинхронно извличане на данни без нарушаване на състоянието на страницата;
- ✓ JQuery, VueJS и други JavaScript библиотеки позволяващи предоставянето на по-качествен потребителски интерфейс и гарантиращи еднакво поведение на системата във всички модерни браузъри;
- ✓ HTML 5 – език за описание на Уеб страници;
- ✓ CSS 3 – език за описание на стилове в Уеб страници;
- ✓ Source code notation – за използване на единен подход при именуване на променливи, имена на класове, функции и процедури. Следи се и за еднотипно подравняване (alignment) на кода с цел по-лесната поддръжка и бързо ориентиране на програмистите при проследяване и отстраняване на несъответствие.
- ✓ Средства за тестване на програмния код – подробно описани по-долу.

Всички изброени технологии представляват последните тенденции в разработката на уеб приложения и се развиват от едни от най-големите и стабилни компании в света, като **Microsoft** (Entity Framework, ASP.NET MVC) и **W3C** (XML, HTML, CSS). Избраните технологии ще гарантират дълга и безпроблемна експлоатация на разработената система, както и лесното ѝ развитие в бъдеще.

7.3.1.2.6. Технологични мерки за постигане на бързодействие

В рамките на реализацията ще бъдат приложени два основни вида мерки за оптимизация и постигане на максимално бързодействие на приложението:

- **Мерки за оптимизация на бързодействието на софтуерното приложение** – набор от добри практики по отношение на разработката на ASP.NET Core приложения, които осигуряват максимално бързодействие на изпълнението на програмния код в приложението
 - **Използване на Inline методи** – при използването на стандартните настройки на компилатора на .NET приложения, относително по-големите методи не биват оптимизирани като inline методи поради по-големия си размер в паметта. С използване на т.нар. aggressive inlining обаче, може да бъде избегнат този проблем, като така реализираните методи се изпълняват по бързо от обикновени методи
 - **Преизползване на конекции (connection pooling)** – по подразбиране .NET приложенията не прилагат преизползване на конекциите към базата. Опитът от работа с .NET Framework обаче показва, че наличието на прекалено много отворени конекции в даден момент води до ненужно натоварване. По тази причина добра практика е да се използва клас, който управлява определен по размер набор от преизползваеми конекции, което води до контролиран брой на конекции, оптимизиран спрямо наличния ресурс.
 - **Кеширане (Caching)** – голяма част от често използваните данни от всички потребители на системата – номенклатури, списъци от потребители, списъци от роли и функционални права, елементи на навигация и т.н., подлежат на кеширане в съответна структура за съхранение на ниво приложение, така че извикването им да не предизвиква постоянно извикване на базата данни – roundtrip.. При предлаганата архитектура с използване на ясно разграничение на слоя на достъп до данни и използване на Repository pattern е лесно да се осигури как кеша винаги да се използва, както и да се осигури че няма как той да не се актуализира при въвеждане на стойност, която трябва да промени състоянието му, тъй като има единна точка за достъп до информацията, чрез съответния repository клас.
 - **Компресиране (Compression)** – платформата позволява настройване на автоматично компресиране на изпращани между сървъра и клиента файлове и данни, което ускорява бързодействието.
 - **Редуциране на заявките (HTTP Request)** – използването на JavaScript базиран интерфейс като Vue.js позволява пълен контрол над това как само тези данни които следва да се презаредят и обменят между клиент и сървър да бъдат изпращани към контролерите. При традиционните модели на MVC приложение това е пълните данни за екрана, но при подхода на single page application, може да се контролира до ниво поле точно кои данни да се изпратят. Това редуцира трафика и ненужното натоварване.
 - **Минифициране на ресурсите** – редуцира размера на използваните JavaScript файлове, което отново намалява трафика и ускорява обмена с брауъра.

- Мерки за оптимизация на бързодействието на базата данни – оптимизация на структурата на базата данни и заявките
 - **Индексиране** – създаването на индекси по колони включени в условия за join на таблици и условия за филтриране на заявки ускорява заявките към базата данни
 - **Разделяне на таблици** – функциите за разделяне на таблици в Microsoft SQL Server позволява елегантно разделяне на големи таблици в отделни под-таблици (child tables), на база на разделяща функция (partitioning function). Това позволява намаляване на редовете, които трябва да обработят заявките и скок в бързодействието, когато заявките сканират данни в определена под-таблици. При подобна оптимизация може да се постигнат разделяния на данните които да осигуряват значително подобрене на извличането на най-често използваните данни.

7.3.1.2.7. Технологични мерки за постигане на информационна сигурност и защита на данните

Системата ще бъде реализирана с изключителен акцент върху сигурността, поради естеството на информацията обработвана в системата и големия обем информация, който се очаква да бъде концентриран с течение на използването. Системата ще отговаря на изискванията на техническото задание като осигури:

- Контрол на достъпа до всички ресурси на системата с изключение на публично достъпните функционалности.
- Функционалността на информационната система, която управлява достъпа до ресурсите ѝ, ще поддържа **област за собственото изпълнение**, защитена от външни въздействия и от опити да се следи хода на работа, поради естеството на модула за управление на достъпа като подсистема отделена от останалата функционалност;
- Информационната система ще разполага с технически и/или програмни средства, позволяващи периодично да се **проверява коректността на компонентите на функционалността** на информационната система, която управлява достъпа до ресурсите ѝ;
- Защитните механизми ще преминат **penetration тест**, който да потвърди, че неоторизиран ползвател няма очевидна възможност да получи достъп до функционалността на информационната система, която управлява достъпа до ресурсите ѝ;
- Функционалността на информационната система, която управлява достъпа до ресурсите ѝ, ще осигури взаимна **изолация на процесите** чрез разделяне на адресните им пространства.
- Ще бъдат взети всички мерки за защита на механизмите за обмен на данни, като всички видове обмен на данни са защитени със използване на **SSL криптиране и клиентски сертификати**.

7.3.1.3. Защита на информацията при изпълнение на проекта

В хода на изпълнение на проекта Сمارт Системс т и неговите служители се задължават:

- да опазват и да не разгласяват пред трети лица съдържанието на документацията, която е станала известна при изпълнението на проекта, без писменото съгласие на Възложителя, с изключение на случаите, когато са задължени по закон за това;
- да опазват и да не разгласяват пред трети лица съдържанието на чувствителна и друга информация, лични данни и друга защитена информация, която е станала известна при изпълнението на проекта;
- да опазват и да не разгласяват пред трети лица информация, която е станала известна при изпълнението на проекта относно вътрешни правила и процедури, структура, начин на функциониране на Възложителя, комуникации, мрежи и информационни системи на Възложителя;
- да спазват вътрешните правила за достъп и режим на работа в сградите на Възложителя;
- да спазват всички процедури и изисквания на Възложителя за работа в информационната инфраструктура.

7.3.1.4. Подход към реализацията на изискванията за информационна сигурност

За да се гарантира сигурността на данните и да се елиминира злоупотребата с лични данни и възможността от подмяна на самоличност се преминава през няколко стъпки включващи размяна на кодове за достъп и токени, които са криптирани и електронно подписани.

Системата ще реализира модел на сигурността базиран на роли и права (role-based access control) за оторизацията на достъпа на потребителите до ресурсите предоставяни от приложението. Моделът на сигурността базиран на роли и права реализиран в проекта ще бъде базиран на предлагания от Microsoft.NET Framework role-based security model. Microsoft.NET подсистемата за управление на достъпа на потребителите е базирана на потребителски роли, с помощта, на които за всеки потребител може да се определи дали има достъп до определен ресурс и дали ма право да извърши дадена операция.

По този начин ще бъдат реализирани изискванията за сигурност към реализацията на системата:

Съхранението и достъпът до данните се осъществяват чрез системи за управление на бази данни. Всички документи, файлове и данни следва да се съхраняват в системи за управление на бази данни.

Системата ще има добре развита концепция и механизъм за регистриране на потребители, определяне на техните права, роли и съответно допустими функции в потребителския интерфейс.

- Системата ще включва комбинация от мерки за сигурност на данните
- Системата ще поддържа следните методи за автентификация и авторизация на потребителите и техните действия със системата:

- автентификация с потребителско име и парола;
- Механизмът за авторизация ще извършва проверка за това дали съществуват необходимите права при всеки опит за достъп до ресурс на Системата. Механизмът за авторизация ще бъде свързан към журнала на достъпа, който представлява регистър на потвърдените и отказани авторизации. Журналът на достъпа ще съдържа минимум следната информация:
 - идентификатор на страната изискваща достъп (потребител/съд, система и др.);
 - IP адрес, от който се достъпва Системата;
 - точно време на осъществяване на достъпа;
 - ресурс, който се достъпва;
 - операция (четене, редакция, запис, изтриване);
 - статус на достъпа.
- Всяко създаване на достъп ще е проследим и за него трябва да се регистрират автоматично и по начин, не подлежащ на корекция или унищожаване, данни за: създател на достъпа, данни, еднозначно идентифициращи служител, време на създаване на достъп, разпореждане за достъпа.
- Системата ще да осигурява сигурен механизъм за запис на всички събития, свързани със сигурността (контрол на достъпа и оторизацията, контрол на системните и конфигурационни промени, спиране и пускане на системата) в реално време. Записите ще дават достатъчно информация за еднозначно идентифициране на потребител извършил действието, време на събитието и извършено действие.
- Системата ще има функционалност, която да изготвя отчет за всички опити за неправомерен достъп до системата.
- Системата ще позволява проследяване и запис на действията на всички потребители (да регистрира в база данни всички потребителски действия: влизане в системата, преглед, търсене, въвеждане, коригиране и изтриване на данни).
- Системата ще осигурява на потребител с административни права създаване на доклади, основани на горепосочените записи, по зададени от него критерии.
- Системата ще има интерфейс за административно уведомяване на активните потребители.
- Системата ще позволява контрол на версиите.

7.3.1.5. Обосновка на избрания подход за реализация на изискванията за информационна сигурност

Изборът на платформа за реализация Microsoft.NET Framework гарантира, че програмният код, реализиращ функционалността на системата ще се изпълнява в защитена среда (sandbox) и ще е изолиран от хардуерната и софтуерната среда, в която работи. Платформата Microsoft.NET притежава вградена защита от атаки от типа на cross site scripting, SQL Injection, buffer overflow и др.

Системата ще поддържа лог информация за осъществения достъп от потребителите приложението. Лог информацията ще съдържа като минимум следната информация:

- Дата, час и секунда на осъществяване на достъпа;
- IP адрес, от който е осъществен достъпа;

- Идентификатор на потребител/съд, система, който е осъществил достъпа;
- Идентификатори на данните, които са били достъпни и вида на извършената операция.

Сигурност – модела MVC позволява въвеждане на всички механизми за сигурност предлагани от .Net Framework – различни методи автентикация, валидация на HTML заявките, отделяне на публична и вътрешна част на функционалността, като за разлика от стандартните уеб приложения, всяко извикване на метод от контролера преминава през авторизация.

Съхранението и достъпът до данните ще се осъществяват чрез системи за управление на бази данни (СУБД). Всички документи, файлове и данни ще се съхраняват в релационна СУБД.

При коригиране на данни, системата ще иска потвърждение от потребителя.

Данните няма да могат физически да се изтриват от системата. На тези данни се поставя флаг „отменени“/ „анулирани“, „с изтекъл срок“, „изтрити“ или „архивирани“.

Достъпът до отделните страници и компоненти на системата ще се контролира чрез система за идентификация на потребителите. Механизмът за оторизиране и идентифициране ще бъде централизиран чрез електронен подпис. Системата ще поддържа подсистеми за автентикация и авторизация на потребителите и техните действия в системата. Те ще поддържат:

- автентикация с потребителско име и парола;
- автентикация чрез интеграция с директорийна услуга;
- автентикация с усъвършенства и/или квалифициран електронен подпис, издаден от доставчик на удостоверителни услуги, съответно - доставчик на квалифицирани удостоверителни услуги;
- автентикация чрез средства за електронна идентичност съгласно Закона за електронната идентификация;
- възможност за конфигуриране на сложността на паролите за вход в Системата – дължина, малки/големи букви, съдържание на определени символи и др.

7.3.1.6. Управление, идентифициране и предоставяне на права на потребители

Имплементацията на модела за сигурност ще Membership provider модел за управление на сигурността, информацията за сесиите и функционалните права. Така разработения Membership provider ще контролира достъпа до всеки метод на контролерите на Системата, като за частите от Системата, които изискват автентикация тя ще може да се реализира с потребителско име и парола.

Административния панел ще включва изглед за управление на потребителите, в който ще могат да добавяни, изтривани, блокирани и редактирани потребители.

Системата ще прилага модел на групово задаване на потребителски права базиран на функционални роли, които биват присвоявани чрез атрибути на класа за Membership provider, така че в приложението ще е налична информация за ролите на текущия потребител, и на база на това ще бъдат активирани и скривани функции от страниците на приложението.

Списъка на потребителите със специални права в административния панел Администратори на системата (супер администратор/администратор) следва да имат следните възможности:

- добавяне, изтриване, редактиране на потребител.
- смяна на парола на потребител
- заключване на потребител
- добавяне, премахване на потребителска роля
- отключване на потребител, чийто профил е блокиран поради неуспешни опити за автентикация

7.3.1.7. Идентификация и предоставяне на права

Електронната идентификация на потребителите в системата ще бъде подчинена на следните изисквания:

- Системата ще поддържа стандартен подход за регистрация на потребители с потребителско в съвъръра за идентификация и управление на достъп;
- Процесът по регистрация на потребители ще бъде максимално опростен и бърз, но ще включва специфични стъпки.
- Визуализиране на информацията относно стъпките по регистрация и информация във връзка с процеса за потвърждаване на регистрацията и активиране на потребителския профил.
- В membership provider на приложенията ще се предвидят мерки за блокиране на достъп при неколнократни неуспешни опити за влизане
- Избор на парола с контекстна валидация на полето (in-line validation) и визуализиране на сложността на паролата като "слаба", "нормална" и "силна";
- В зависимост от ролята, потребителите ще получават съответни права.
- Ще бъде реализирана интеграция с националната схема за електронна идентификация съгласно изискванията на Закона за електронната идентификация и действащите нормативни правила за оперативна съвместимост. За целта подсистемата за автентикация и оторизация на потребителите ще поддържа интеграция с външен доставчик на идентичност.

7.3.1.8. Подход за реализация на ETL процеса за миграция на данни

7.3.1.8.1. Аргументация на подхода за реализация на ETL процеса

При избора на методиката за реализация на ETL процес следва да се избере една от многото алтернативни възможности за реализация на процеса: алтернативи по отношение на използваните средства, по отношение на структурата на процеса, по отношение на моделирането на мета данните.

Първия избор при започване на реализацията на ETL процеса се отнася до това дали да се реализира отделно приложение, в което да се програмира фиксирано в програмния код логиката за изпълнение на процеса по миграция на данни, да се реализира процеса изцяло със средствата на СУБД, чрез съхранени SQL процедури, или реализация чрез специфичен продукт за реализация на ETL процеси и интеграция на данни. С цел максимална гъвкавост и възможността за по-ефективен трансфер на знания към екипа на Възложителя, подходящият избор е използване на готов продукт за реализация на ETL – в случая продукта интегриран в предлаганото от нас СУБД, за което НСИ притежава лиценз (Microsoft SQL Server), а именно -SQL Server Integration Services.

Втория избор при реализацията на ETL процеса е в избора на начина за извършване на трансформацията на данните – при подхода ETL, трансформацията се извършва преди зареждането на данни в многомерната база данни (extract -> transform -> load), при алтернативния подход се използват средствата на самата база данни, така че суровите данни се зареждат в хранилището и се трансформират на място в него (extract -> load -> transform). Използването на SQL Server Integration Services като средство за дефиниране на ETL позволява дефинирането на параметризиран и конфигуриран процес на реализация на ETL.

Допълнителни технологични решения включени в предлагания подход са използването на област за съхраняване на сурови (оригинални) данни, както и наличието на staging database. От гледна точка на пълната проследимост на процеса (audit trail), от гледна точка на по-сигурното управление на оригиналните данни (поддържането им в контролирана от системата област осигурява по-добър контрол от ограничаването до съхраняване на пътища до файловата система), както и от гледна точка на по-доброто разграничаване на етапите от процеса (с наличието на staging база се отделя процеса на извличане от процеса на самата трансформация), използването на тези два допълнителни компонента ще доведе до по-стабилна и устойчива архитектура.

7.3.1.8.2. Предлаган подход за реализация на ETL за миграция на данни

Предлагания процес за реализация на ETL за миграция на данни от съществуващата база данни Informix в надградената версия на системата, на база данни Microsoft SQL Server включва следните основни стъпки:

1. Анализ на структурата и механизмите на съществуващия модел на данни – във фазата на проектиране на ETL процеса, ще се извърши задълбочен анализ на съществуващия модел на данни, използваните системни процедури, тригери и функции, структурата и консистентността на данните. На база този анализ ще се проектира съответстващ модел на данни в СУБД SQL Server и ще се специфицират нужните разработки за реализация на модела при новата СУБД.
2. Разработка на модел на данни в SQL Server на ИСБТ – чрез адаптация и надграждане на съществуващия модел на данни в Informix към спецификата на СУБД Microsoft SQL Server, разработка на staging модел на данни в SQL Server използван за огледален трансфер на данни от Informix и разработка на механизми за мапинг на staging данните към новия модел на данни, включително обработка, изчистване, елиминирание на дубликати и тн.
3. Извличане на данните от източниците – концепцията на миграцията приема, че ще се създаде staging база със структура идентична с тази на Informix базата данни на текущата версия на системата. Чрез ETL процес, базиран на пакет на SSIS, чрез наличните ODBC драйвери за достъп до СУБД Informix, се извършва извличане на данни и прехвърляне на записите в staging база.
4. Стъпка на валидиране и изчистване на данните – в тази стъпка се извършват операции свързани с осигуряване на интегритета на данните и тяхната коректност. Извършва валидация на данните, както и анализ на отклоненията в данните, налагат се проверки на ограничения и връзки между обектите.
5. Трансформиране на данните – при тази стъпка заредените в staging база данни, ще бъдат трансформирани в модела за данни на надградената версия база данни на ИСБТ (т.нар. Logical Data Map).

7.3.1.8.3. Инструмент за реализация на ETL процеса – SQL Server Integration Services

За реализацията на ETL процеса предлагаме SQL Server Integration Services, включени във функционалностите на SQL Server Standard Edition, които предлагат нужния инструментариум за изграждане на зареждане на данни от разнообразни видове СУБД, конфигурируеми трансформации на данните, както и възможност за визуално моделиране.

Основното предназначение на SQL Server Integration Services (SSIS) е чрез т.нар. трансформационни пакети да предложи средство за извличане на данни от разнообразни източници (релационни бази и файлове в различни формати като текст, CSV, XML), трансформацията им, съобразно бизнес логиката на пакета и зареждане в бази данни. По своето същество SQL Server Integration Services (SSIS) е платформа за изграждане на решения за data integration и изграждане на ETL процес за зареждане на хранилища от данни с високо бързодействие и ниво на параметризация и лесна поддръжка.

Основните предимства на предлагания инструмент за ETL са:

1. Бърз и ефективен начин за разработка на ETL процеса, поради използването на готовите възможности на средството, както и поради наличието на визуални редактори за дефиниране на трансформациите и стъпките на процеса
2. Богати възможности за настройване на изпълнението и поведението при грешки
3. Оптимизирано бързодействие чрез отделяне на контролния процес от процесите на зареждане на отделни набори данни и използване на паралелизация
4. Лесно преизползване на функционалността и разширяване без промяна в програмен код или SQL процедури

7.3.1.8.4. Предлаган инструмент за реализация на OLAP кубове – SQL Server Analysis Services

SQL Server Analysis Services е средството за изграждане на OLAP кубове интегрирано в СУБД Microsoft SQL Server.

SQL Server Analysis Services поддържа OLAP кубове, като за разлика от други OLAP инструменти се поддържат не един, а всички концептуални подходи за реализация на кубове, което дава максимална гъвкавост при изграждане на многомерния модел:

- ROLAP – Relational OLAP, при който данните се съхраняват в релационен модел и не се съхраняват в базата данни с многомерен модел.
- MOLAP – Multidimensional OLAP, при който данните се извличат от релационен модел и се съхраняват в многомерен модел
- HOLAP – Hybrid OLAP, при който основните данни са в релационния модел, но част от предварителните агрегации и индекси се съхраняват в многомерната база за оптимизация на бързодействието

SQL Server Analysis Services поддържа множество алтернативи за изграждане на заявките към OLAP кубовете:

- MDX – специализиран език за заявки към многомерни кубове
- SQL – лимитирана възможност за стандартни заявки към многомерните данни
- DMX – език за заявки към модели за data mining
- LINQ – възможност за изграждане на композитни, динамично структурирани заявки към кубовете от .NET приложения, елиминираща нуждата от работа с фиксирани SQL заявки и елиминиращо нуждата от кодиране на бизнес логика в слоя на базата данни.

SQL Server Analysis Services е естествено интегрирано в предлаганата база данни, в която по идентичен начин е интегрирано и предлаганото ETL средство за реализация на системата. SQL Server Analysis Services в комбинация с базата данни на SQL Server напълно

отговаря на изискванията на техническото задание за изграждане и поддържане на аналитична база данни:

- Дефиниране на многомерна база данни;
 - Дефиниране на дименсии в многомерна БД;
 - Дефиниране на кубове с агрегирани данни.
 - Извличане на данни от релационни и нерелационни източници на данни;
 - Агрегиране на извлечените данни;
 - Зареждане на агрегирани данни в многомерна БД;
 - Съхраняване на многомерна БД като таблици в релационна СУБД;
 - Създаване на OLAP-модели;
 - Наличие на графичен интерфейс за работа с OLAP-структурите;
 - Притежава система за архивиране и възстановяване на данни и експортиране в ASCII файл;
 - Извършване на многомерен анализ върху многомерни БД;
 - Притежава набор от визуални и аналитични средства, осигуряващи възможност за избор, представяне, манипулиране, визуализация и анализ на данните;
 - Притежава средство за генериране на отчети (Analysis Services се интегрира в рамките на платформата с SQL Server Reporting services);
 - Притежава административно средство за управление на потребители, групи от потребители и връзки към базата данни (SQL Management Studio Express).
- Всички посочени възможности се базират на наличния лиценз притежаван от Възложителя за SQL Server Standard Edition 2016 и по нов.

7.3.2. Функционални възможности на основните компоненти на реализацията

Предлаганата архитектура на системата предполага строго спазване на принципите на обектно ориентираното проектиране и програмиране и компонентно базираната архитектура, която е в сърцето на обектно-ориентирания подход. От тази гледна точка системата ще реализира всички компоненти специфицирани в заданието на системата (категоризирани като основни модули на системата), но базирани на ядро от стандартизирана функционалност за обработка на статистическа информация в системата. Функционалността на ядрото се представя под формата на услуги, които се извикват от реализация на функционалности в крайните компоненти, но е реализирана на едно място в ядрото и е достатъчно генерична, за да може да бъде извиквана и от другите компоненти на системата. Компонентите на надградената система ще покриват всички функционални възможности описани в т. 4.1. от Техническото задание, специфицираща наличните

средства в текущо използваната версия на системата

7.3.2.1. Компонент „Събиране на данни за външнотърговски потоци“

Компонентът ще предоставя функционалност за зареждане на първичните микроданни за външнотърговската дейност на страната, постъпващи по трите канала на информация (ЕКСТРАСТАТ, ИНТРАСТАТ и Статистически наблюдения), в изградените Първични регистри на наблюдения в БД на системата. Функционалността ще се реализира, като се използва стандартизирания модул за импорт и въвеждане на данни на ядрото на системата, който съдържа в себе си възможност за описание с мета данни на форматите за импорт на данни, правилата за валидиране на данни, както и начина по който събраните данни се съотнасят към модела на данни на първичните регистри. Функционалността поддържа и нужните връзки, така че събраните данни да се асоциират към правилните респонденти и данните за популацията, за която се регистрира наблюдението.

7.3.2.2. Компонент „Редактиране и обработка на микроданни“

Компонентът предоставя потребителски интерфейс, чрез който технолозите на микроданни и методолозите търсят, извличат, визуализират и редактират записи с микроданни от Първичните регистри на наблюдения, както в режим на единична така и в режим на пакетна обработка на данни. В системата съхранява пълна история на извършените промени в записи с първични микроданни и позволява указани от потребителя промени, за които има записи в историята на промените, да бъдат сторнирани и да бъдат възстановени оригиналните стойности на съответните полета. Компонентът предоставя и функционалност, чрез която се извършва автоматично коригиране на някои проблеми в първичните данни, свързани с не попълнени задължителни полета и некоректни стойности в някои от полетата. Тези автоматични корекции се извършват по предварително дефинирани алгоритми и се прилагат върху първичните данни за ЕАД и редовете от декларациите по ИНТРАСТАТ. Посочените автоматични корекции и техните алгоритми са ще бъдат реализирани чрез дефиниране на мета данни за условна обработка и функционалността на модул „Логика на обработка на първичните данни“. Компонентът поддържа и цялата функционалност за работа с Крайните регистри на наблюдения при

успешното обработване на постъпили първични микроданни.

7.3.2.3. Компонент „Актуализиране на данни за търговците от СОРТ“

Компонентът осигурява на вътрешен регистър на данни за търговците, които са необходими за функционирането на системата и са съхранени в данни-структури, оптимизирани за нуждите на системата. В съответствие с изискванията, част от тези данни (като например регистрационните данни) се извличат от СОРТ, а друга част (като например данни за оборотите на търговците) се изчисляват от системата на база на постъпилите първични данни за външнотърговските потоци. Интеграцията се извършва на база на стандартизирания интеграционен интерфейс, който е част от ядрото на системата.

7.3.2.4. Компонент „Предоставяне на данни на заинтересовани потребители“

Компонентът предоставя функционалност за автоматизирано генериране на данни-файлове за външни потребители, с които НСИ има сключено споразумение или договореност, както и за вътрешни потребители от НСИ. Тези изходни файлове се генерират от Крайните регистри на наблюдения, като се използва функционалността на стандартизиран модул за експорт на данни, част от ядрото от функционалност на системата. Експортните формати се дефинират с мета данни, така че евентуално променяне на структурата на форматите да не води до промени в програмния код. Системата събира информация за извършваните експорти за външни източници и техния обхват.

7.3.2.5. Компонент „Поддържане на общосистемна информация“

Компонентът осигурява въвеждане и поддържане на използваните класификации, номенклатури и конфигурационни данни за системата, както и добавяне на нови класификации и номенклатури при нужда. Ядрото на системата позволява поддържане на пълно версионирание на класификационните данни и номенклатури, управление на тяхната валидност и преизчисляване на данните в съответствие с промените в тях.

7.3.3. Добавена нова и променена функционалност в надградената версия на системата

След разработване, тестване и одобряване от възложителя на функционалността на системата към новата технологична платформа и изпълнение и одобряване на резултатите от миграцията на данните от СУБД Informix към СУБД Microsoft SQL Server, актуализираната версия на системата ще бъде надградена чрез добавяне на нови функционалности и промяна на съществуващи такива.

- Ще бъдат реализирани следните промени и допълнения при зареждане, редактиране и обработване на първични микроданни:
 - Промени в структурата и съдържанието на входни файлове за зареждане на първични микроданни.
 - Промени и допълнения в правилата за формален и логически контрол при обработване на първични микроданни.
 - Промени и допълнения в правилата за автоматични корекции на първични микроданни.
 - Добавяне на функционалност за автоматично анулиране на определени записи с първични микроданни.
 - Добавяне на нови полета в първични регистри.
 - Добавяне на нови таблици с конфигурационни данни, необходими за параметризиране на някои обработки и експорти на данни.
 - Реализиране на допълнителни функционалности в потребителския интерфейс на екраните за редактиране и обработване на първични микроданни.
 - Реализиране на възможности за търсене по допълнителни полета в историята на промените в данните от Първичните регистри.
- Ще бъдат реализирани следните в алгоритмите за изготвяне на оценки на не отговорили и не обхванати потоци:
 - Реализиране на промени в алгоритъма за изчисляване на оценки на не обхванати потоци.

- Реализиране на промени в алгоритъма за изчисляване на оценки на не отговорили търговци.
- Ще бъдат реализирани следните промени в структурата, формата на първичните регистри, както и във функционалностите и начина на зареждане на „Регистъра на търговците“ в ИСВТ:
 - Разширяване на „Регистъра на търговците“ с допълнителни бизнес характеристики на търговците.
 - Реализиране на допълнителни функционалности в потребителския интерфейс на „Регистъра на търговците“ с възможност за автоматично зареждане от ИСБС и РСЕ на бизнес характеристиките и други данни за търговците.
 - Потребителският интерфейс да предоставя възможност за ръчно въвеждане на липсващи данни за търговците, с изключение на данните за оборотите им и на задълженията им по Интрастат.
 - Добавяне на актуални контакти за търговците в Данни за търговците в полето „Забележка“.
 - Реализиране на допълнителни функционалности в потребителския интерфейс на „Регистъра на търговците“ с възможност за директно зареждане в „Регистъра на търговците“ на получаваните от НАП файлове с данни за регистрирани и де регистрирани по ДДС лица и за задълженията на търговците по Интрастат.
- Ще бъдат реализирани следните промени в Класификацията на Митническите учреждения

Промените ще са свързани с кардинални промени на класификацията като съдържание (наименованията на Митническите учреждения) и йерархия (нова структура на Митническите учреждения). Те ще позволяват кодът на митническото учреждение да не е уникален, а да е възможно дублиране при затворен срок на валидност на съществуващия код (по подобие на въвеждането на нови лимити за една и съща стока). Уникална ще остане комбинацията от код и интервал на валидност. Няма да бъдат предвиждани припокривания на сроковете на валидност.

Форматът и срока на валидност ще се промени от ММ–ГГГГ на ДД–ММ–ГГГГ.

- Ще бъдат реализирани промени и допълнения във всички съществуващи Изходни данни-файлове, генерирани от ИСВТ за външни и вътрешни потребители.
- Ще се извърши добавяне на нови данни-файлове, генерирани от ИСВТ за външни и вътрешни потребители.
 - Данни-файлове от крайните регистри с контролни суми по Интрастат и Екстрастат в детайлизация по източници.
 - Данни-файлове, необходими за изготвяне на доклад за качеството.
 - Данни-файлове, необходими за определяне на праговете.
 - Данни-файлове за метаданните за обмена на микроданни с станалите държави членки
 - Данни-файлове, необходими за Енергийните баланси.
 - Данни-файлове по 3 знака на КПИД, необходими за отдел „Нефинансови национални и регионални сметки“.
 - Данни-файлове за електрическа енергия.
 - Данни-файлове за МЗХГ.
 - Данни-файлове с разлики между стойностите, декларирани от търговците по Интрастат и по ДДС.
 - Данни-файлове за Евростат по валута на фактуриране.
- Ще се добави функционалност за изчисляване на индекси на цени на износа и вноса на стоки и за съставяне на Каталог на стоки представители на износа и вноса.
- Ще се добави функционалност за изчисляване на коефициенти за транспортни разходи и застраховки, необходими за изчисляване на търгуваните обеми по цени CIF и FOB.

- Ще се добави функционалност за производство на данни по бизнес характеристики на предприятията с определяне на конфиденциалните данни.
- Ще се добави функционалност за зареждане на данни от ДДС декларации, получени от другите държави членки (VIES).
- Ще се създаде нова БД „MDE” за нуждите на обмена на микроданни с останалите държави членки.
- Ще се създадат допълнителни функционалности за експортиране и съхраняване в БД „MDE” на микроданните, необходими за обмена с останалите държави членки, съобразно последните изисквания за структура, формат и съдържание на полетата.
- Ще се създаде допълнителна функционалност за генериране на файлове от БД „MDE” с микроданните, необходими за обмена с останалите държави членки, съобразно последните изисквания за структура, формат и съдържание на полетата.
- Ще се създаде допълнителна функционалност за импортиране в БД „MDE” на микроданните, получени при обмена от останалите държави членки.
- Ще се създаде допълнителна функционалност в ИСВТ за автоматично коригиране, логически контрол, обработване, съхраняване и анализ в БД „MDE” на микроданните, получени при обмена от останалите държави членки.

7.3.4. Архивиране и проследяване на промените

Смарт Системс 2010 ЕООД ще реализира системен процес, който да извършва ежедневен бекъп на системата, да следи правилната работа на компонентите (health check), да събира информация за параметрите на бързодействието на компонентите. В случай на грешка при изпълнението на ежедневния бекъп, администраторите на системата и Смарт Системс т ще бъдат уведомени по електронна поща, след което екипът на Смарт Системс ще извърши ръчно бекъпа с използване на команда `pg_dump dbname > outfile`.

Ежедневния бекъп извършван от автоматизирания процес ще работи на принципа на Continuous Archiving and Point-in-Time Recovery (PITR) – този начин за архивиране се базира на създаване на базов бекъп (full backup), който се използва като начална точка на верига от състояния на базата, към които системата може да бъде възстановена във всеки момент. След създаване на базовия архив се архивират само transaction log файловете, които съхраняват лог на транзакциите на системата. По този начин бекъпа може да се изпълнява максимално често и без да се изразходва дисково пространство за пълен бекъп, а поради възможността за максимална честота на бекъпите се елиминира потенциалната възможност за загуба на данните от последния бекъп, която съществува при нормален dump на базата, който не може да се изпълнява толкова често.

Проследяването на промените се извършва чрез имплементиране на механизми за автоматично съхраняване на историята на промените на всички обекти в слоя за достъп до данните на системата, реализиран на база на Entity Framework Core. За целта ще бъдат прихващани събитията по прилагане (submit changes/commit) на транзакциите към базата данни извиквани от функционалността на приложението, след което с рефлексия се идентифицират променените полета и записи от данните и се съхранява предишното състояние и промененото състояние на стойностите им. Така регистрираната информация се съхранява под формата на пълен лог на промените в данните за всички таблици от модела и може да бъде използвана като пълно функционален audit trail.

Базите данни на SQL Server които ще се използват в системата, ще бъдат с Full Recovery режим на работа, който позволява Point-in-Time Recovery (PITR) при нужда от възстановяване на базата до предишно нейно състояние.

7.3.4.1.1. Подход за резервното копиране и възстановяване

За реализиране на изискванията към резервното копиране и възстановяване, предлагаме следния подход:

Автоматизирано създаване на архивните копия, реализирано чрез средствата предлагани от базата данни Microsoft SQL Server.

За поддържането на доброто състояние на базата данни и създаването на архивни копия ще

бъде създаден съответния Maintenance Plan в Microsoft SQL Server. Добра доказана практика за архивиране на данните е архивирането да се извършва на следните 2 нива:

- Transaction Log Backup – архивиране на разликите, извършва се на всеки час;
- Full Backup – пълен архив, извършва се на всеки 24 часа.

Посоченият модел на архивиране позволява високо ниво на сигурност на данните и минимизира времето за възстановяване на системата при необходимост.

Microsoft SQL Server поддържа архивирането на базата данни да се извършва on-line, като това не се отразява на работещите с базата данни системи.

7.3.4.1.2. Процедура за възстановяване на данните

За възстановяване на данните процедурата е следната:

При неработоспособна система:

1. Възстановява се работоспособността на операционната система (ако е необходимо), като се използва системно резервно копие на операционната система и инсталираните приложения или се инсталира отново операционната система и необходимите приложения;
2. Възстановява се системата за управление на база данни (ако е необходимо);
3. Възстановяват се данните от последното направено резервно копие, както и прилежащите към него инкрементални резервни копия;
4. Провеждат се тестове за работоспособност на системата;
5. Отбелязва се извършването на възстановяването в дневник на администратора, като се отбелязват: датата и часът на възстановяването, системният администратор/и, извършил възстановяването и резултати от тестването на системата.

При работоспособна система:

1. Възстановяват се данните от последното направено резервно копие, както и прилежащите към него инкрементални резервни копия;
2. Провеждат се тестове за работоспособност на системата;
3. Отбелязва се извършването на възстановяването в дневник на администратора, като се отбелязват: датата и часът на възстановяването, системният администратор/и, извършил възстановяването и резултати от тестването на системата.

7.4. Изпълнение на специфичните изисквания към дейностите

7.4.1. Дейност 1 Изготвяне на системен проект

7.4.1.1. Описание на дейността

Ще бъде изготвен Системен проект, включващ детайлизация на процесите, функционалностите и обхвата на системата и интеграцията ѝ с други системи, процеси, дейности и нормативни изисквания идентифицирани по време на анализа, модела на цялостната система и отделните модули, електронните форми, включително полета, съдържанието и валидацията им, модел на потребителски интерфейс, езици, на които се предоставят данните и интерфейса, план график за разработка и план за миграция на данни (ETL);

7.4.1.2. Изпълнение на дейността „Изготвяне на системен проект“

При изпълнението на тази дейност, в съответствие с посочените в т. 6.2. от Техническото задание специфични изисквания за изготвяне на Системен проект, ще бъде включена информация за:

- Определяне на концепция за реализация на информационната система на базата на техническото задание и резултатите от анализите от предходния етап;
- Дефиниране на детайлни изисквания и бизнес процеси, които трябва да се реализират в ИСВТ – полета, контроли и валидация, форми, справки;
- Дизайн и модел на електронните форми, включително полета, съдържанието и валидацията им съобразени със установените актуални стандарти на ДАЕУ;
- Детайлен план за мигрирането данните на ИСВТ и съхраняването им в единна база данни (ETL процес), включващо като минимум определяне на източниците на данни, осигуряване на качеството на данните, нормализация и изчистване на данните, съответствието им към новия модел на данните в ИСВТ, подход за набавянето на липсващи данни, верификация и оптимизация с цел бързодействие съгласно установените добри практики;
- Определяне на потребителския интерфейс - създаване и одобрение на прототип на потребителския интерфейс на системата;
- Изготвяне на план график, съобразен с поредността залегнала в анализа за техническа реализация по модули, процеси и функционалности на ИСВТ.

Изготвеният Системен проект трябва да бъде одобрен от Възложителя, в срок от 10 работни дни. В случай на забележки, корекции или допълнения от страна на Възложителя, Сمارт Системс ще ги отрази в системния проект в срок не по-късно от 10 работни дни.

7.4.1.3. Резултати

В резултат на изпълнение на дейност 1 „Изготвяне на системен проект“ ще бъде изготвен и одобрен Системен проект с детайлизация на функционалностите на

системата и интеграцията ѝ с другите системи, модела на електронните форми, включително полета, съдържанието и валидацията им, модел на потребителски интерфейс, план график за разработка и план за миграция на данни (ETL).

7.4.2. Дейност 2 Развитие на информационна система „Външна търговия“ за миграция на платформата на модул „Микроданни“ чрез използването на съвременни високотехнологични ИТ решения.

7.4.2.1. Описание на дейността

Ще бъдат разработени всички модули на ИСВТ, съгласно изискванията на настоящото техническо задание, идентифицираните процеси, обхват и функционалности, залегнали в одобрения Системен проект. Ще бъдат проведени вътрешни контролни тестове на ИСВТ; Ще се изготвят детайлни сценарии за провеждане на приемателните, функционални и интеграционни тестове на ИСВТ.

7.4.2.2. Изпълнение на дейността

Изпълнението на дейност 2 ще започне с оборудване на работна среда и анализ и уточняване на изискванията към миграцията на модул „Микроданни“ като ще включи следните под дейности:

- Мигриране на релационната БД на модул „Микроданни“ от Informix към нов продукт:
 - Мигриране на данни-модела на релационната БД на модул „Микроданни“ от Informix към нов продукт;
 - Мигриране на програмните модули в БД (stored procedures, functions & triggers);
 - Мигриране на конфигурационните скриптове, използвани за зареждане на метаданни и класификации;
 - Вътрешно тестване на миграцията на програмни модули и метаданни в БД.
- Създаване на потребителско приложение на модул „Микроданни“ чрез използване на нов продукт:
 - Създаване и модернизиране на потребителското приложение чрез нов продукт;
 - Вътрешно тестване на потребителското приложение на модул „Микроданни“.
- Техническа миграция на данните от модул „Микроданни“
- Инсталиране и конфигуриране на сървър на модул „Микроданни“
 - Създаване и конфигуриране на машина във виртуална среда;
 - Инсталиране и конфигуриране на сървър на БД на новия продукт;
 - Инсталиране и конфигуриране на Web-сървър за нуждите на новия продукт.
- Инсталиране и конфигуриране на тестови сървър на модул „Микроданни“
 - Създаване и конфигуриране на машина във виртуална среда;
 - Инсталиране и конфигуриране на тестови сървър на БД на новия продукт;
 - Инсталиране и конфигуриране на Web-сървър за нуждите на новия продукт;
 - Актуализиране на техническата документация на модул „Микроданни“

- Приемателно тестване на миграцията на модул „Микроданни“
 - Разработване на скриптове и процедури за миграция на микроданните от БД Informix към новия продукт;
 - Създаване на схемите в тестовата БД и инсталиране на потребителското приложение в тестовата среда;
 - Миграция на данните от експлоатационната БД Informix към тестовата БД на новия продукт;
 - Провеждане на приемателно тестване на мигрираната модул „Микроданни“

7.4.2.3. Резултати

В резултат на изпълнение на дейност 2, ще бъдат изготвени приемателни тестове, съгласно които НСИ ще проведе приемно тестване на системата и ще бъде разработена и функционираща мигрирана ИСВТ.

7.4.3. Дейност 3 Развитие на информационна система „Външна търговия“ за добавяне на нови функционалности и промяна на съществуващите.

7.4.3.1. Описание на дейността

Изпълнението на дейност 3, включва добавяне на нови функционалности и промяна на съществуващите.

7.4.3.2. Изпълнение на дейността

Изпълнението на дейност 3 включва реализирането на функционални промени и допълнения в модул „Микроданни“. Изпълнението ѝ ще започне с анализ и уточняване на изискванията и включва следното:

- Промени и допълнения при зареждане, редактиране и обработване на първични микроданни:
 - Промени в структурата и съдържанието на входни файлове за зареждане на първични микроданни;
 - Промени и допълнения в правилата за формален и логически контрол при обработване на първични микроданни;
 - Промени и допълнения в правилата за автоматични корекции на първични микроданни;
 - Добавяне на функционалност за автоматично анулиране на определени записи с първични микроданни;
 - Добавяне на нови полета в първични регистри;
 - Добавяне на нови таблици с конфигурационни данни, необходими за параметризиране на някои обработки и експорти на данни;

- Реализиране на допълнителни функционалности в потребителския интерфейс на екраните за редактиране и обработване на първични микроданни;
- Реализиране на възможности за търсене по допълнителни полета в историята на промените в данните от Първичните регистри;
- Промени в алгоритмите за изготвяне на оценки на не отговорили и не обхванати потоци:
 - Реализиране на промени в алгоритъма за изчисляване на оценки на не обхванати потоци;
 - Реализиране на промени в алгоритъма за изчисляване на оценки на не отговорили търговци.
- Промени в структурата, формата на първичните регистри, както и във функционалностите и начина на зареждане на „Регистъра на търговците“ в ИСВТ: - Разширяване на „Регистъра на търговците“ с допълнителни бизнес характеристики на търговците.
 - Реализиране на допълнителни функционалности в потребителския интерфейс на „Регистъра на търговците“ с възможност за автоматично зареждане от ИСБС и РСЕ на бизнес характеристиките и други данни за търговците.
 - Потребителският интерфейс да предоставя възможност за ръчно въвеждане на липсващи данни за търговците, с изключение на данните за оборотите им и на задълженията им по Интрастат.
 - Добавяне на актуални контакти за търговците в Данни за търговците в полето „Забележка“
 - Реализиране на допълнителни функционалности в потребителския интерфейс на „Регистъра на търговците“ с възможност за директно зареждане в „Регистъра на търговците“ на получаваните от НАП файлове с данни за регистрирани и де регистрирани по ДДС лица и за задълженията на търговците по Интрастат.
- Промени в Класификацията на Митническите учреждения
 - Промените са свързани с кардинални промени на класификацията като съдържание (наименованията на Митническите учреждения) и йерархия (нова структура на Митническите учреждения). Промените трябва да позволяват кодът на митническото учреждение да не е уникален, а да е възможно дублиране при затворен срок на валидност на съществуващия код (по подобие на въвеждането на нови лимити за една и съща стока). Уникална остава комбинацията от код и интервал на валидност. Не се предвижда припокриване на сроковете на валидност.
 - Освен това, форматът на срока на валидност трябва да се промени от ММ-ГГГГ на ДД-ММ-ГГГГ.

- Промени и допълнения във всички съществуващи Изходни данни-файлове, генерирани от ИСВТ за външни и вътрешни потребители.
- Добавяне на нови данни-файлове, генерирани от ИСВТ за външни и вътрешни потребители:
 - Данни-файлове от крайните регистри с контролни суми по Интрастат и Екстрастат в детайлизация по източници;
 - Данни-файлове, необходими за изготвяне на доклад за качеството;
 - Данни-файлове, необходими за определяне на праговете;
 - Данни-файлове за метаданните за обмена на микроданни с останалите държави членки;
 - Данни-файлове, необходими за Енергийните баланси;
 - Данни-файлове по 3 знака на КПИД, необходими за отдел „Нефинансови национални и регионални сметки“;
 - Данни-файлове за електрическа енергия;
 - Данни-файлове за МЗХГ.
 - Данни-файлове с разлики между стойностите, декларирани от търговците по Интрастат и по ДДС.
 - Данни-файлове за Евростат по валута на фактуриране.
- Добавяне на функционалност за изчисляване на индекси на цени на износа и вноса на стоки и за съставяне на Каталог на стоки представители на износа и вноса.
- Добавяне на функционалност за изчисляване на коефициенти за транспортни разходи и застраховки, необходими за изчисляване на търгуваните обеми по цени CIF и FOB.
- Добавяне на функционалност за производство на данни по бизнес характеристики на предприятията с определяне на конфиденциалните данни.
- Добавяне на функционалност за зареждане на данни от ДДС декларации, получени от другите държави членки (VIES).
- Създаване на нова БД „MDE” за нуждите на обмена на микроданни с останалите държави членки.
- Създаване на допълнителни функционалности за експортиране и съхраняване в БД „MDE” на микроданните, необходими за обмена с останалите държави членки, съобразно последните изисквания за структура, формат и съдържание на полетата.
- Създаване на функционалност за генериране на файлове от БД „MDE” с микроданните, необходими за обмена с останалите държави членки, съобразно последните изисквания за структура, формат и съдържание на полетата.
- Създаване на допълнителна функционалност за импортиране в БД „MDE” на микроданните, получени при обмена от останалите държави членки.
- Създаване на допълнителна функционалност в ИСВТ за автоматично коригиране, логически контрол, обработване, съхраняване и анализ в БД „MDE” на микроданните, получени при обмена от останалите държави членки.

7.4.3.3. Резултати

Резултат на изпълнение на дейност 3 е Работеща ИСВТ с внедрени нови функционалности.

7.4.4. Дейност 4 Тестване на системата

7.4.4.1. Описание на дейността

Изпълнението на дейност 4, включва инсталиране на ИСВТ в тестовата среда на Възложителя и провеждане на приемателни, интеграционни и тестове за производителност.

7.4.4.2. Изисквания към изпълнение на дейността

Изпълнението на дейност 4 включва изпълнение на следните дейности:

- Провеждане на тестване на софтуерното решение в създадена за целта тестова среда, за да демонстрира, че изискванията са изпълнени;
- Тестовата среда ще бъде запазена като база за предоставяне на външни изпълнители при адаптиране на системите им към автоматизиран обмен чрез web services;
- Смарт Системс ще предложи и опише методология за тестване, която ще използва в план за тестване с описание на обхвата на тестването, вид и спецификация на тестовите, управление на дефектите, регресионна политика, инструменти, логистично осигуряване и други параметри на процеса.

В резултата на изпълнение на дейност 4, Смарт Системс ще изготви доклад с резултатите от тестването, със заключения и препоръки за подготовката на ИС за внедряване в редовна експлоатация и осигуряване на нейната сигурна и надеждна работа.

Докладът с резултатите от тестването подлежи на одобрение от Възложителя в срок до 10 работни дни. В случай на забележки, корекции или допълнения от страна на Възложителя Смарт Системс ще ги отрази в срок не по-късно от 10 работни дни.

7.4.4.3. Резултати

В резултат на изпълнение на дейност 4 ще бъдат изпълнени следните дейности:

- Инсталирана ИСВТ в тестовата среда на Възложителя;
- Проведени приемателни, интеграционни и тестове за производителност на ИСВТ;
- Изготвен доклад с резултатите от тестването, със заключения и препоръки за подготовката на ИСВТ за внедряване в редовна експлоатация.

7.4.5. Дейност 5 Обучение

7.4.5.1. Описание на дейността

Смарт Системс ще организира и проведе обучение на различните видове групи ползватели и администратори на ИСВТ, на нейните модули и функционалности.

7.4.5.2. Изпълнение на дейността

Смарт Системс ще изготви методология, учебен план и учебни материали.

Смарт Системс ще организира и ще проведе обучения за следните групи и ползватели на софтуерното решение:

- IT служители на Възложителя за придобиване на компетенции за ежедневната поддръжка и администрирането на ИСВТ;
- Служители на НСИ от отдел „Външна търговия“ за работа с ИСВТ, по модули и функционалности.

Смарт Системс ще опише подробно подхода, който ще приложи за постигане на максимална ефективност на обучението, както и за отчитане на резултатите от него.

Смарт Системс ще изготви детайлни програми за обучение описващи съдържанието и продължителността на отделните модули на системата, тестове за оценка на обучаемите и методика за оценка на ефективността и нуждата от допълнително обучение, които ще бъдат одобрени от Възложителя, минимум 10 дни преди провеждане на обучението.

Смарт Системс ще изготви доклад с резултатите от обучението, препоръки за бъдещо развитие на уменията на потребителите, както и методика за обучение на ново постъпили служители в бъдеще. Доклада от обучение подлежи на одобрение от Възложителя в срок от 10 работни дни. В случай на забележки, корекции или допълнения от страна на Възложителя, Смарт Системс ще ги отрази в срок не по-късно от 10 работни дни..

7.4.5.3. Резултати

В резултат на изпълнение на дейност 5 „Обучение“ ще бъдат изпълнени следните дейности:

- Обучени администратори на ИСВТ;
- Обучени потребители от НСИ (статистики) за всички модули на ИСВТ;
- Предоставени всички необходими ръководства за администриране и работа със ИСВТ;
- Одобрена методика за обучение на ново постъпили служители;
- Одобрен доклад с резултати от обучението.

7.4.6. Дейност 6 Внедряване

7.4.6.1. Описание на дейността

Смарт Системс ще организира и извърши инсталиране на ИСВТ в продукционната среда на Възложителя и интеграция с предвидените в Системния проект информационни системи, зареждане на данните в продукционната среда и стартиране на ИСВТ в реална експлоатация, както и следните дейности:

- Инсталиране и конфигуриране на виртуална машина;
- Инсталиране и конфигуриране на операционна система върху виртуалната машина;
- Инсталиране и конфигуриране на уеб сървър върху операционната система;
- Инсталиране и конфигуриране на база данни върху операционната система;
- Инсталиране и конфигуриране на приложен софтуер върху операционната система;
- Осигуряване на три годишна гаранционна поддръжка на изградената операционна среда.

7.4.6.2. Изпълнение на дейността

В изпълнение на дейност 6, Смарт Системс ще изпълни следните изисквания към внедряването:

- Виртуалната машина ще бъде съвместима с виртуалната среда на НСИ;
- Виртуалната среда на НСИ ще може да управлява виртуалната машина с пълния си набор от функционалности;
- Операционната система ще бъде безплатна или НСИ ще има лиценз за нея;
- Операционната система ще има поне три години дългосрочна поддръжка от производителя след въвеждане на ИСВТ в реална експлоатация;
- Операционната система ще актуализира автоматично всички подобрения на сигурността си;
- Операционната система ще бъде защитена от зловреден софтуер;
- Уеб сървърът ще бъде безплатен или НСИ ще има лиценз за него;
- Уеб сървърът ще има поне три години дългосрочна поддръжка от производителя след въвеждане на ИСВТ в реална експлоатация;
- Базата данни ще е безплатна или НСИ ще има лиценз за нея;
- Базата данни ще има поне три години дългосрочна поддръжка от производителя след въвеждане на ИСВТ в реална експлоатация;
- Базата данни ще се поддържа от всички СУСИ и приложения използвани от НСИ за информационни системи;
- Приложният софтуер ще е безплатен или НСИ ще има лиценз за него;
- Приложният софтуер ще има поне три години дългосрочна поддръжка от производителя след въвеждане на ИСВТ в реална експлоатация;
- Приложният софтуер ще се поддържа от всички СУСИ и приложения използвани от НСИ за информационни системи;

- Смарт Системс ще осигури три годишна гаранционна поддръжка на виртуалната машина, операционната система, базата данни и приложния софтуер след въвеждане на ИСВТ в реална експлоатация;
- Гаранционната поддръжка ще включва обновяване на виртуалната машина, операционната система, базата данни и приложния софтуер;
- Смарт Системс ще внедри софтуерното решение в информационната и комуникационна среда на НСИ. Това включва инсталиране, конфигуриране и настройка на програмните и системни компоненти на ИСВТ в условията на експлоатационната среда на зареждане на данните в продукционната среда в съответствие с одобрения План за зареждане на данните в продукционна среда;
- Преди стартиране на редовната експлоатация на системата, задължително ще се проведат всички предвидени обучения по т.6.6 от Техническото задание;
- При стартиране на реалната експлоатация на ИСВТ, Смарт Системс ще осигури минимум 5 свои специалисти в продължение на минимум 30 работни дни, които да оказват незабавна експертна помощ на служителите в процеса на ежедневната им работа, анализират и отстраняват възникнали проблеми и несъответствия в процеса на експлоатация;
- Смарт Системс ще изготви План за бъдещо развитие на ИСВТ, в който да предложи нови модули, функции и възможности за подобрения, ползата и/или нуждата, от които е станала известна след одобрението на Системния проект;
- Смарт Системс ще поеме пълна отговорност за доставката, инсталацията и привеждане в работно състояние на Системата, включително консултации по инсталацията и настройката на хардуера и софтуера, необходим за системата;
- Системата ще бъде разработена така, че да работи с голяма производителност и висока надеждност, като осигурява нормалния технологичен процес, предоставя инструменти за архивиране, запазване на резервни копия и възстановяване от срив без загуба на информация и технологично време;
- Смарт Системс ще изготви План за бъдещо развитие на ИС, в който да предложи нови модули, функции и възможности за подобрения, ползата и/или нуждата от които е станала известна след одобрението на Системния проект.

7.4.6.3. Резултати

В резултат на изпълнение на дейност 6 „Внедряване“ ще бъдат изпълнени следните дейности:

- Инсталирана и настроена ИСВТ в продукционна среда.
- Заредени предварително верифицирани данни в продукционна среда.
- Внедрена в реална експлоатация система.
- Одобрен протокол за внедряване на ИСВТ.

7.4.7. Дейност 7 Гаранционно поддържане

7.4.7.1. Описание на дейността

Смарт системс 2010 ЕООД ще осигури гаранционно поддържане 36 месеца след приемане на ИСВТ в реална експлоатация.

7.4.7.2. Изисквания към изпълнение на дейността

При необходимост и по време на гаранционния период ще бъдат осъществявани дейности по осигуряване на експлоатационната годност на софтуера и ефективното му използване от Възложителя, в случай че настъпят явни отклонения от нормалните експлоатационни характеристики, заложи в Системния проект.

Смарт Системс ще предоставя надлежно услугите по гаранционна поддръжка, като предоставя за своя сметка единна точка за достъп за приемане на телефонни и e-mail съобщения в собствен ИТIL базиран Център за поддръжка, включително достъп на Възложителя до тикет системата на Смарт Системс за преглед и анализ статусите и изпълнението на докладваните тикети.

Приоритетите на проблемите се определят от Възложителя в зависимост от влиянието им върху работата на администрацията. Редът на отстраняване на проблемите се определя в зависимост от техния приоритет.

Минималният обхват на поддръжката ще включва:

- Извършване на диагностика на докладван проблем с цел осигуряване на правилното функциониране на системите и модулите;
- Отстраняване на дефектите, открити в софтуерните модули, които са модифицирани или разработени в обхвата на проекта;
- Консултации за разрешаване на проблеми по предложената от Смарт Системс конфигурация на средата (операционна система, база данни, middleware, хардуер и мрежи), използвана от приложението, включително промени в конфигурацията на софтуерната инфраструктура на мястото на инсталация;
- Възстановяването на системата и данните при евентуален срив на системата, както и коригирането им в следствие на грешки в системата;
- Експертни консултации по телефон и електронна поща за системните администратори на Възложителя за идентифициране на дефекти или грешки в софтуера;
- Актуализация и предаване на нова версия на документацията на системата при установени явни несъответствия с фактически реализираните функционалности, както и в случаите, в които са извършени действия по отстраняване на дефекти и грешки, в рамките на гаранционната поддръжка.
- Промяна на конфигурациите за реализирани регистри и услуги, в случай че го изисква промяна в нормативната уредба;

- По време на гаранционния период на софтуера, всички актуализации при наличие на грешки и промени трябва да бъдат доставени в най-кратки срокове.

Услугите по гаранционна поддръжка ще бъдат достъпни в рамките на стандартното работно време на Възложителя – в работни дни, от 9:00 до 17:30 ч.

През целия период на гаранционната поддръжка, Сمارт Системс ще поддържа системата работоспособна и актуална, като отразява промените в нормативната база, в срока, в който е предвидено те да влязат в сила.

В периода на поддръжката Смарт Системс ще актуализира Плана за бъдещо развитие на ИСВТ с нови модули функции и възможности за подобрене, ползата от които е станала известна след стартиране на редовната експлоатация.

Дейността завършва с подписване на Протокол за извършена гаранционна поддръжка на ИСВТ в редовна експлоатация. Дейността ще бъде извършена за периода, посочен в неговото предложение за изпълнение на поръчката, 36 месеца след приемане на ИСВТ в реална експлоатация.

7.4.7.3. Резултати

В резултат на изпълнение на дейност 7 „Гаранционно поддържане“ ще бъдат изпълнени следните дейности:

- Актуална и работоспособна ИСВТ;
- План за бъдещо развитие на ИСВТ;
- Извършена гаранционна поддръжка на ИСВТ.

8. ДОКУМЕНТАЦИЯ

Предлагаме да изготвим документация за улесняване ползването и функционирането на системата и сме включили разходите за това в ценовото предложение.

8.1. Изисквания към документацията

Цялата документация и всички технически описания, ръководства за работа, администриране и поддръжка на Системата, включително и на нейните съставни части, ще бъдат налични и на български език.

Всички документи ще бъдат предоставени от „Смарт Системс 2010“ ЕООД в електронен формат (ODF/Office Open XML/MS Word DOC/RTF/PDF/HTML или др.), позволяващ пълно текстово търсене/търсене по ключови думи и копиране на части от съдържанието от оригиналните документи във външни документи, за вътрешна употреба на възложителя.

Навсякъде, където в документацията има включени диаграми или графики, те ще бъдат вградени в документите в оригиналния си векторен формат.

„Смарт Системс 2010“ ЕООД ще предостави на Възложителя следната документация:

- Детайлна техническа документация на програмния приложен интерфейс (API), включително за поддържаните уеб услуги, команди, структури от данни и др. Документацията ще бъде придружена и с примерен програмен код и/или библиотеки (SDK) за реализиране на интеграция с външни системи, разработен(и) на Java или .NET. Примерният код ще бъде напълно работоспособен и ще демонстрира базови итерации с API-то:
 - Регистриране на крайна точка (end-point) за получаване на актуализации от Системата в реално време;
 - Заявки за получаване на номенклатурни данни (списъци, таксономии);
 - Заявки за актуализиране на номенклатурни данни (списъци, таксономии);
 - Регистрация на потребител;
 - Идентификация и оторизация на потребител или уеб услуга.
- Документацията за приложния програмен интерфейс (API) ще бъде публично достъпна;
- Всеки предоставен REST приложно-програмен интерфейс ще бъде документиран чрез API Blueprint (<https://github.com/apiaryio/api-blueprint>),

Swagger (<http://swagger.io>) или чрез аналогична технология. Аналогично представяне ще бъде изготвено и за SOAP интерфейсите;

- Детайлна техническа документация за схемата на базата данни – структури за данни, индекси, дялове, съхранени процедури, конфигурации за репликация на данни и др.
- Ръководства на потребителя и администратора за работа и администриране на Системата
- Обща информация, инструкции и процедури за администриране и поддръжка на приложните сървъри, сървърите за бази данни и др.
- Обща информация, инструкции и процедури за администриране, архивиране и възстановяване, и поддръжка на сървъра за управление на бази данни.

8.1.1. Методология за управление на проектната документация

Прилагането на адекватна методология за управление на документите е един от най-важните аспекти при управление на качеството на ИТ проектите. Методологията за управление на документите цели да се регламентират дейностите и отговорностите по разработката, утвърждаването, разпределението, актуализирането, съхранението и архивирането на документите в проекта. В конкретния проект ще бъде приложена Методология за управление на документите съгласно RUP, а така също и образците по RUP, когато за изискваните по проекта документи са предвидени такива.

Структура и съдържание на документите по RUP

Съгласно методологията RUP документите по проекта, ще се изготвят със следната структура:

<Заглавна страница>
Контрол на документа
Подписи
История на промените
Разпространение на документа
Съдържание
<Заглавие – автоматично генерирано от полето ‘Title’>
1. Въведение
1.1. Предназначение на документа
1.2. Цел и обхват
1.3. Приложимост
1.4. (Отклонения)
1.5. Абревиатури, съкращения и дефиниции
1.6. Позовавания и приложими документи
1.6.1. Позовавания
1.6.2. Приложими документи
<Глави, специфични за съответния документ>

Цикъл на живот. Статуси

Съгласно RUP и Техническата спецификация на проекта, за обозначаване на статуса на отчетните резултати през отделните фази и итерации на проекта ще се използват следните символи:

E – (Evolving) В процес на разработка

S – (Signed off) Приет (утвърден, подписан от възложителя)

R – (Revised) Ревизиран (актуализиран след приемане, при необходимост)

C – (Complete) Завършен (окончателен, документа не подлежи на по нататъшна промяна)

Конвенция за именуване на файлове по RUP

Примерна конвенция за именуване на файлове:

<Код на документа>-<Дата>-<Код на проекта>-(-Т-)<Версия>(<R>)-<Статус>-<Език>

- Код на документа - състои се от 3 букви по стандарта за кодиране;
- Дата - формат на датата YYYYMMDD, където Y-година, M-месец, D-ден;
- Код на проекта = ECS, за всички вътрешни документи на Проекта;
- Т се включва по показания начин за индикация на документ – образец;
- Версия – формат на версията X.Y., където Y е номер на регистрирана промяна на документа, а X е номер на официалното издание. Y не се нулира.
- R (ревизирана версия) се появява след номера на версията тогава, когато е прегледана версия и са нанесени бележки и/или промени с ‘track changes’.
- Статус – статус на документа:
 - E – (Evolving) В процес на разработване;
 - S – (Signed off) Одобрен;
 - R – (Revised) Променен след като е одобрен;
 - C – (Complete) Финален.
- Език – език на документа =BG(български), EN(английски).

8.2. Прозрачност и отчетност

В обхвата на проекта е включено извършване на дейности по анализ на бизнес процеси и нормативна уредба, проектиране на системна и приложна архитектура, разработване на компютърни програми и други дейности, свързани с предоставяне на

специализирани професионални услуги. Сمارт Системс и Възложителят ще публикуват подробни месечни отчети в машинно-четим отворен формат за извършените дейности, включително количеството изработени човеко-дни по дейности, извършени от консултанти, експерти, специалисти и служители на Смарт Системс и Възложителя.

Документацията, предоставена от Смарт Системс на възложителя, ще бъде:

- на български език;
- на хартия и в електронен формат; копирането и редактирането на предоставените документи следва да бъде лесно осъществимо;
- актуализирана в съответствие със съгласувана с възложителя процедура, която ще включва документи, подлежащи на промяна/актуализация, крайни срокове и нужната за случая методология.

Минимално изискуемата документация по проекта включва долу изброените документи.

8.3. Системен проект

Смарт Системс ще дефинира в детайли конкретния обхват на реализация на софтуерната разработка и ще документира изискванията към софтуера в детайлна техническа спецификация (системен проект), която ще послужи за пряка изходна база за разработка.

При документирането на изискванията, с цел постигане на яснота и стандартизация на документите, ще се използва утвърдена нотация за описание на бизнес модели. Изготвената детайлна техническа спецификация (системен проект) се представя за одобрение на възложителя. В случай на забележки, корекции или допълнения от страна на възложителя Смарт Системс е длъжен да ги отрази в детайлната техническа спецификация (системен проект).

8.4. Техническа документация

Всички продукти, които ще се доставят, ще бъдат със специфична документация за инсталиране и/или техническа документация, в това число:

- Ръководство за администратора, включващо всички необходими процедури и скриптове по инсталиране, конфигуриране, архивиране, възстановяване и други, необходими за администриране на системата;
- Документи за крайния ползвател – Смарт Системс т ще предостави главното Ръководство на ползвателите на софтуера. Документът е предназначен за крайните ползватели. Той трябва да описва цялостната функционалност на приложния софтуер и съответното му използване от крайни ползватели;
- Детайлно описание на базата данни;
- Описание на софтуерните модули;

- Описание на изходния програмен код.

8.5. Протоколи

Смарт Системс ще изготвя протоколи от изпълнението на различните етапи на проекта, заедно със съпътстващите ги документи – резултати от изпълнението на етапите.

8.6. Комуникация и доклади

За успешното изпълнение на проекта и съгласно техническото задание, „Смарт Системс 2010“ ЕООД предлага използването на доказан механизъм за управление на проектната комуникация, който е неразделна част от предлаганата цялостна проектна методология.

Управлението на комуникацията ще включва изготвяне на минимум следните регулярни доклади за статуса и напредъка на изпълнението на поръчката:

8.6.1. Встъпителен доклад

Встъпителният доклад трябва ще бъде предоставен до един месец от подписването на договора и ще съдържа описание минимум на:

- Подробен работен план и актуализиран времеви график за периода на проекта;
- Начини на комуникация;
- Отговорни лица и екипи.

Встъпителният доклад следва да бъде одобрен от възложителя.

8.6.2. Междинни доклади

Междинните доклади ще бъдат представяни и ще се предават при приключване на всяка от дейностите и под-дейностите и/или при настъпване на събитие.

Междинните доклади ще съдържат информация относно изпълнението на дейностите и под-дейностите по предварително изготвения проектен план.

Докладът за междинния напредък ще бъде подготвен по следния начин:

- Общ прогрес по дейностите през периода;
- Постигнати проектни резултати за периода;
- Срещнати проблеми, причини и мерки, предприети за преодоляването им;
- Рискове за изпълнение на свързани дейности и на проекта като цяло и предприети мерки;
- Актуализиран план за изпълнение, ако има такъв.

Всеки междинен доклад следва да бъде одобрен от възложителя.

8.6.3. Окончателен доклад

В края на периода за изпълнение ще бъде представен окончателен доклад. Окончателният доклад ще съдържа описание на изпълнението и резултати.

Докладите се изпращат до отговорния служител на възложителя. За тази цел възложителят ще определи в договора отговорния/отговорните служител/служители. Всички доклади се представят на български език в електронен формат и на хартиен носител. Докладите се одобряват от отговорния/отговорните служител/служители в срок до 5 работни дни.

Всички доклади ще се представят на Възложителя на български език - на хартиен и на електронен носител. Представянето на докладите ще се извършва чрез подписване на двустранни предавателно-приемателни протоколи, подписани от представители на Смарт Системс и на Възложителя.

Възложителят разглежда представените доклади и уведомява Смарт Системс за приемането им без забележки или ги връща за преработване, допълване и/или окомплектоване, ако не отговарят на изискванията, като чрез упълномощено в договора лице дава указания и определя срок за отстраняване на констатираните недостатъци и пропуски.

9. РЕЗУЛТАТИ

С изпълнението на настоящата обществена поръчка се очаква да бъдат постигнати следните основни резултати:

Разработена, въведена в експлоатация и реално функционираща Информационна система „Външна търговия“. Обучени служители за работа със системата..

София, 15.11.2019 г.

Подпис:

/Марин Кошутов –

Управител

Смарт системс 2010 ЕООД/