

до

ПРЕДСЕДАТЕЛЯ НА
ВЪРХОВНИЯ КАСАЦИОНЕН СЪД
ГР.СОФИЯ, БУЛ. „ВИТОША“ №2

ТЕХНИЧЕСКО ПРЕДЛОЖЕНИЕ

за участие в обществена поръчка чрез публична покана по реда на Глава осма „а“ от ЗОП с предмет „Оптимизация на бързодействието на Автоматизираната система за управление на делата АСУД и интегриране с единната среда за безхартиен обмен на дела във Върховния касационен съд“

От Сمارт системс 2010 ЕООД

със седалище и адрес на управление София 1408, ж.к. ИВАН ВАЗОВ, бл. 69, ет. 5, ап. 20

регистрирано в Търговския регистър на Агенция по вписванията с ЕИК 201020584,

представявано от Марин Виктор Кошутков,

в качеството си на Управител

УВАЖАЕМИ ГОСПОДИН ПРЕДСЕДАТЕЛ,

След запознаване с всички документи и образци от документацията за участие, удостоверяваме и потвърждаваме, че фирмата ни отговаря на посочените изисквания и условия. Същото доказваме с документите, приложени към настоящото Техническо предложение:

1. Предлагаме да изпълним поръчката при следните условия:

Изпълнителят ще извърши всички описани изисквания на Техническата спецификация, използвайки методология RUP за разработката на проекта, и предлагайки технологичен подход за реализация на проекта, описан детайлно в приложените документи :

- План за изпълнение на проекта
- Предложение за оптимизация на модела на данни
- Предложение за интеграция на АСУД с единна среда за безхартиен обмен на дела на ВСС
- План за тестване
- Техническо описание на разработеното решение

(участникът описва подробно своето предложение (концепция), съобразено с Техническата спецификация на Възложителя, като добавя толкова редове, колкото е необходимо)

2. Срок за изпълнение – 3 месеца, считано от датата на подписване на договор, както следва:

- 2.1. Анализ и проектиране, изграждане на прототип на системата –1 месец;
- 2.2. Тестова експлоатация на прототипа в продуктивна среда –1 месец;
- 2.3. Окончателно внедряване - 1 месец.

3. Декларирам, че ще се запазят всички данни и документи в системата.

4. Декларирам, че промените в системата ще се извършват без спиране работата на АСУД във ВКС и в работно време.

Съгласно

5. За изпълнение на предмета на поръчката не се изисква закупуване на допълнителен или приложен софтуер.

6. Към офертата прилагаме Ценово предложение и подписан от нас проект на договор.

7. Срок на валидност на предложението – считаме се обвързани от условията, задълженията и отговорностите, поети с направеното от нас предложение и приложенията към него, представляващи негово съдържание 30 (тридесет) календарни дни от крайния срок за получаване на офертите.

8. При изпълнението на поръчката ще се придържаме точно към указанията на Възложителя, съгласно документацията за участие.

9. При сключването на договор ще представим необходимите документи, издадени от компетентен орган, за удостоверяване липсата на обстоятелствата по чл.47 ал.1 т.1 и декларация за липса на обстоятелствата по чл.47 ал.5 от Закона за обществените поръчки.

Приложения:

- План за изпълнение на проекта
- Предложение за оптимизация на модела на данни
- Предложение за интеграция на АСУД с единна среда за безхартиен обмен на дела на ВСС
- План за тестване
- Техническо описание на разработеното решение

Подпис:

Дата: 07/12/2015

Име и фамилия: Марин Кошутков

Длъжност: Управител

**ТЕХНИЧЕСКО ПРЕДЛОЖЕНИЕ
ЗА ИЗПЪЛНЕНИЕ НА ПОРЪЧКАТА**

от

„Смарт системс 2010“ ЕООД

**за изпълнение на обществена поръчка по реда на Глава осма „а“ от ЗОП с предмет
„Оптимизация на бързодействието на Автоматизираната система за управление на
делата АСУД и интегриране с единната среда за безхартиен обмен на дела във
Върховния касационен съд“**

УВАЖАЕМИ ДАМИ И ГОСПОДА,

С представянето на настоящето предложение, заявяваме нашето желание, ако бъдем определени за изпълнител на обществената поръчка с горепосочения предмет да я изпълним добросъвестно, професионално, качествено и в срок.

I. Декларираме, че сме запознати с техническата спецификация и условията за участие в обявената от Вас обществена поръчка за избор на изпълнител и гарантираме, че сме в състояние да изпълним качествено поръчката, в пълно съответствие с действащото законодателство, изискванията на Възложителя и представения проект на договор.

II. Нашето конкретното предложение за изпълнение на поръчката е, както следва:

СЪДЪРЖАНИЕ НА ТЕХНИЧЕСКОТО ПРЕДЛОЖЕНИЕ:

1.	ВЪВЕДЕНИЕ	3
2.	ПРЕДСТАВЯНЕ НА СМАРТ СИСТЕМС 2010 ЕООД.....	3
3.	КОНЦЕПЦИЯ ЗА РЕАЛИЗАЦИЯ НА ПРОЕКТА	8
2.1.	МЕТОДОЛОГИИ	8
	<i>Методология RUP</i>	<i>8</i>
	<i>Методология за софтуерна разработка.....</i>	<i>12</i>
	<i>Език за програмиране.....</i>	<i>14</i>
	<i>Развойна среда.....</i>	<i>14</i>
2.2.	УПРАВЛЕНИЕ НА ПРОМЕНИТЕ.....	15
2.3.	УПРАВЛЕНИЕ И ОТСТРАНЯВАНЕ НА УСТАНОВЕНИТЕ НЕСЪОТВЕТСТВИЯ В СОФТУЕРНАТА РАЗРАБОТКА	19
2.4.	УПРАВЛЕНИЕ НА КАЧЕСТВОТО	24
	<i>Методика за управление на качеството</i>	<i>24</i>
	<i>Планиране управлението на качеството</i>	<i>24</i>
	<i>Системата по управление на качеството</i>	<i>25</i>
	<i>Целите на системата по управление на качеството са:.....</i>	<i>25</i>
	<i>Средства за управление на качеството:.....</i>	<i>26</i>
	<i>Осигуряване на качеството</i>	<i>26</i>
	<i>Управление на конфигурацията:</i>	<i>27</i>
	<i>Тестване 28</i>	
2.5.	УПРАВЛЕНИЕ НА РИСКА.....	30
	<i>Методология за управление на риска.....</i>	<i>31</i>
	<i>Идентифициране на рисковете</i>	<i>33</i>



1. Въведение

Настоящият документ представя нашето виждане за изпълнението на проекта, като се дефинират методологиите за работа и разработка, управление на риска, управление на проекта, управление на промените.

2. Представяне на Сمارт системс 2010 ЕООД

Смарт системс 2010 ЕООД е сред основните компании в областта на внедряването и разработката на информационните системи в българската съдебната система.

Създадена през 2010 година, 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

Предмет на дейност: „Смарт системс 2010“ ЕООД е сред основните компании в областта на внедряването и разработката на информационните системи в българската съдебната система. Компанията извършва:

- Бизнес консултиране (управление и оптимизация на бизнес процеси, бизнес моделиране и анализ);
- Изготвяне на тръжни документации за кандидатстване по грантови схеми и публично-частни партньорства;
- Управление на проекти и изготвяне на отчети;
- Интеграция на приложения;
- Системна интеграция

Използваните технологии в „Смарт системс 2010“ ЕООД са:

- Microsoft – Visual Studio .Net, ASP, MS SQL, MS IIS;
- IBM Lotus Domino;

Бизнес фокусът на фирмата е насочен основно върху консултиране, разработка, внедряване и обучение на информационни системи.

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.

Създадена през 2010 година, 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.

Екип: Екипът на „Смарт системс 2010“ ЕООД се състои от талантиливи и мотивирани специалисти, посветени на своята работа.

Ние сформираме екипите в зависимост от целите на своите клиенти. Едно от условията за приемане в екипа на „Смарт системс 2010“ ЕООД е кандидатите да притежават талант и желание за развитие, да бъдат посветени на работата си.

Нашият екип се развива постоянно и работи с водещите световни технологии, които отговарят на очакванията на нашите клиенти. Той е съставен от професионалисти от различни области, обединени от целите на фирмата и устремени към оптимално изпълнение на решенията ѝ.

Стратегически проекти: По-важните проекти на „Смарт системс 2010“ ЕООД са:

Проект: Автоматизирана система за управление на делата в съдилищата (АСУД)

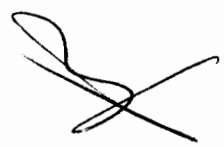
Комплексна система за автоматизация на управление на съдебни на българските съдилища на всички нива.

Проектът покрива нуждите от автоматизиране на процесите по събиране, управление и анализ на информацията по граждански, наказателни, административни, търговски и фирмени дела в съдилищата, наблюдение на процеса по обработка на делата от получаване на входящи документи до поддръжката на база данни с приключените дела.

Проектът отговаря на специфичните нужди за развитие на компетентна управленска структура чрез организирането на обучения на съответните служители на от съдебната система.

Основните модули на разработената система са:

- Регистрационен модул, който позволява документите да бъдат регистрирани в електронни регистри дефинирани от приложението;
- Управление на сканирани документи;
- Модул отговарящ за начина на управление на работните процеси;
- Модул за генериране на документи базирани на формуляри;
- Модул за известяване, използван за извеждане на напомняния след като дадено събитие настъпи;

- 
- Възможности за интеграция с електронна поща;
 - Възможност за електронно подписване;
 - Модул за съхранение, отговорен за съхранението и управлението на всякакъв вид документи;
 - Архивиращ модул;
 - Административен модул;
 - Модул за представяне на информация от делата в Интернет и управление на съдържанието и изграждане на динамични страници за съдилища, хоствани на централен уеб сървър.

Проект: Електронен портал за достъп до информация от съдебни дела и електронни услуги

Електронният портал за достъп до информация от съдебни дела и електронни услуги е тясно интегриран с АСУД и **функционира успешно в 15 съдилища използващи АСУД**, като позволява на регистрираните в него адвокати да имат единна точка за достъп до свързана информация по дела от всички съдилища, в които те се явяват като представители по дела. За разлика от други имплементирани решения в тази насока електронният портал е реализиран на база на електронни услуги и стандартите за обмен на данни между системите за управление на дела използвани в съдебната система, като поради това портала може да бъде интегриран със системите на различни съдилища независимо от използваната технология при спазване на споменатите стандарти за реализация на обмена на данни.

Електронният портал дава възможност за прецизирано даване на достъп до информация, като всеки адвокат получава достъп до делата след като в информационната система на съда се отрази участието му по дадено дело. Достъпната информация включва:

- Обща информация за делото (вх.номер, номер на дело, вид на дело, предмет)
- Информация за датите на съдебни заседания, резултата от тях
- Съдебни протоколи в оригинален вид (без заличаване на лични данни), конвертирани в PDF формат
- Съдебни актове в оригинален вид
- Сканирани документи приложени в делото
- Призовки и съобщения генерирани по делото

Електронният портал има първата реализирана цялостна функционалност за реализация на електронно призоваване, при която електронният портал получава информация от издадените в системата за управление на дела призовки, автоматично ги публикува за достъп от адвокати и други потребители, а в допълнение може да бъде активирана функция която връща електронно потвърждение подписано с електронен подпис за постъпване на съобщението в портала, както и при потвърждаване на получаване на съобщението.

Проект: Разработване на модул за анотиране на съдебни актове и предоставяне на разширено търсене на съдебни актове за ВКС

Посочения модул е интегриран в имплементацията на АСУД реализирана за ВКС, като позволява след изготвяне на съдебните актове, съдебни помощници да анотират съдебните актове с одобрени от изготвилите акта магистрат ключови думи. Така анотираните актове са достъпни от речник на ключовите думи както и чрез форма за разширено търсене в която

могат да се дефинират комбинирани критерии чрез използване на ключови думи, пълнотекстово търсене, посочване на период на издаване на акта, отделение, предмет на делото и т.н.

Проект: Интеграция на АСУД и Деловодна информационна система (ДИС) за Военноапелативен съд с Единната информационна система за противодействие на престъпността

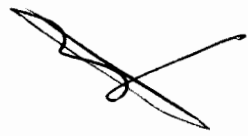
Единната информационна система за противодействие на престъпността е ключов проект в рамките на стратегията за реформа в съдебната система на България чрез имплементиране на информационни технологии. Системата се базира на централна компонента – база данни събираща пълната информация за всяко наказателно производство на базата на информация изпратена от ведомствените информационни системи на всички институции свързани с него. Екипът на „Смарт системс 2010“ ЕООД взе участие в посочения проект извършвайки интеграцията на две информационни системи за управление на дела – АСУД и ДИС, чрез разработка на комуникационни адаптори и конвертори на данни, които позволяват извличане на данни от тези системи и филтриране и трансформиране на тези данни в стандартизирани XML съобщения за възникнали събития от наказателното производство, изпращани за регистрация в централната компонента.

Проект: Интеграция на АСУД с Информационната система за производствата по несъстоятелност (ИСПН) към Министерство на правосъдието

Информационната система за производствата по несъстоятелност (ИСПН) е централизирана информационна система поддържана от Министерство на правосъдието, събираща данни от всички системи за управление на дела използвани в съдебната система на България, по отношение на пълния ход на разглежданите в тях производства по несъстоятелност. ИСПН е разработена като обобщаваща база данни, приемаща информация чрез електронна уеб услуга, предоставяща функционалност за подаване на данни, описани в специален на стандарт за XML описание на информацията за делата, страните, заседанията и постановените актове по дела за несъстоятелност. Екипът на „Смарт системс 2010“ ЕООД, разработи механизми за синхронно и асинхронно изпращане на информация към ИСПН при въвеждането и в АСУД, както и първоначална миграция на информацията за наличните незавършили производства по несъстоятелност.

Експертиза в областта на правосъдието

Екипът на „Смарт системс 2010“ ЕООД е разработил и поддържа Унифицираната информационна система (УИС) използвана от Прокуратурата на РБ, използвайки платформата Метафор 2.0 и специфични възможности на системата за управление на документите. По време на проекта ние сме предоставили услуги за административна оптимизация на процесите; Balance Score Card; Enterprise Architecture и развитие на ИТ стратегия, и разработихме множество специфични функции на управлението на документи, които да покриват широк спектър от дейностите на прокуратурата. Отделно от УИС и услугите реализирани към нея, ние осигурихме централно ниво на държавната администрация и местното самоуправление и общините с персонализирани DMS приложения и интеграция с моделите на бизнес процеси.



Силни страни и предимства:

Предимствата на "Смарт системс 2010" ЕООД са опита в изпълнението на сходни проекти, както и специфични познания, експертиза и квалифицирани експерти. Основните ни предимства са:

- Познаване на европейската и национална рамка за електронно управление;
- Опит в анализ и реинженеринг на работни процеси от целия проектен цикъл свързан с разработката и внедряването на електронни административни услуги и информационни системи за електронното управление;
- Опит в осигуряването на оперативна съвместимост и информационна сигурност на разработените информационни системи;
- Опит в изготвянето на заявления за вписване на информационни обекти в съответните регистри;
- Опит в предоставяне на консултантски услуги, проектиране, разработка, внедряване, обслужване, поддръжка и развитие на ИТ решения;
- Осъществяване бизнес и системен анализ, изготвяне на функционални и технически спецификации;
- Провеждане на обучения за работа с информационни системи;
- Осигуряване на гаранционна и следгаранционна поддръжка на разработения софтуер.

Към настоящия момента „Смарт системс 2010“ ЕООД участва в изпълнението на следния проект във Висш съдебен съвет: **„Разработване и внедряване на единен портал на електронното правосъдие, и обучение за работа с единния портал“**, като **уеб базираното приложение, предмет на разработка на настоящата поръчка, трябва да се интегрира в единния портал на електронното правосъдие.**

В допълнение, „Смарт системс 2010“ ЕООД поддържа системите за управление на съдебните дела АСУД във Върховния касационен съд, в Софийски апелативен съд и АСНС, в окръжни съдилища - ОС Благоевград, ОС Смолян, ОС Велико Търново, ОС Кърджали и ОС Монтана и в следните районни съдилища – РС Благоевград, РС Петрич, РС Гоце Делчев, РС Разлог, РС Смолян, РС Девин, РС Мадан, РС Златоград, РС Чепеларе, СРС, както и системата СУСД в СРС.

Системата АСУД е разработена по проект на ААМР и е интелектуална собственост на ВСС. Екипът на „Смарт системс 2010“ ЕООД включва двамата основни разработчици на системата от самото и създаване през 2001 година, извършили всички актуализации, интеграции и разширявания на функционалността.

Системата СУСД е разработена по проект на Министерство на правосъдието през 2006 г.. В момента най-големия съд използващ тази система е СРС, като фирма „Смарт системс 2010“ поддържа базата данни и администрира приложението на СУСД в СРС, като фирмата успешно е извършвала два договора за оптимизация на бързодействието в тази система.

Този факт е предимство за участника предвид факта, че единният портал на ВСС, предмет на настоящата поръчка, трябва да се интегрира със системите за управление на съдебните дела, като „Смарт системс 2010“ ЕООД добре познава и поддържа 2 от тях.



3. Концепция за реализация на проекта

За управление на софтуерната разработка за изпълнение на настоящата обществена поръчка ще се прилага методологията RUP (Rational Unified Process), както и добри практики и принципи на методологията PMBOK (Project Management Body Of Knowledge).

2.1. Методологии

Предлагаме при практическото приложение на концепцията за реализация на проекта, да се използва RUP - една от най-добрите методологии и инструментариуми за прилагането на изложения по-горе подход при реализацията на ИТ проекти.

Методология RUP

RUP е итеративен и инкрементален процес, който позволява непрекъсната еволюция и подобрение 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 подпроекти с ясна последователност от дейности с определен план и оценителни критерии, завършващи с артефакт за бенефициента.

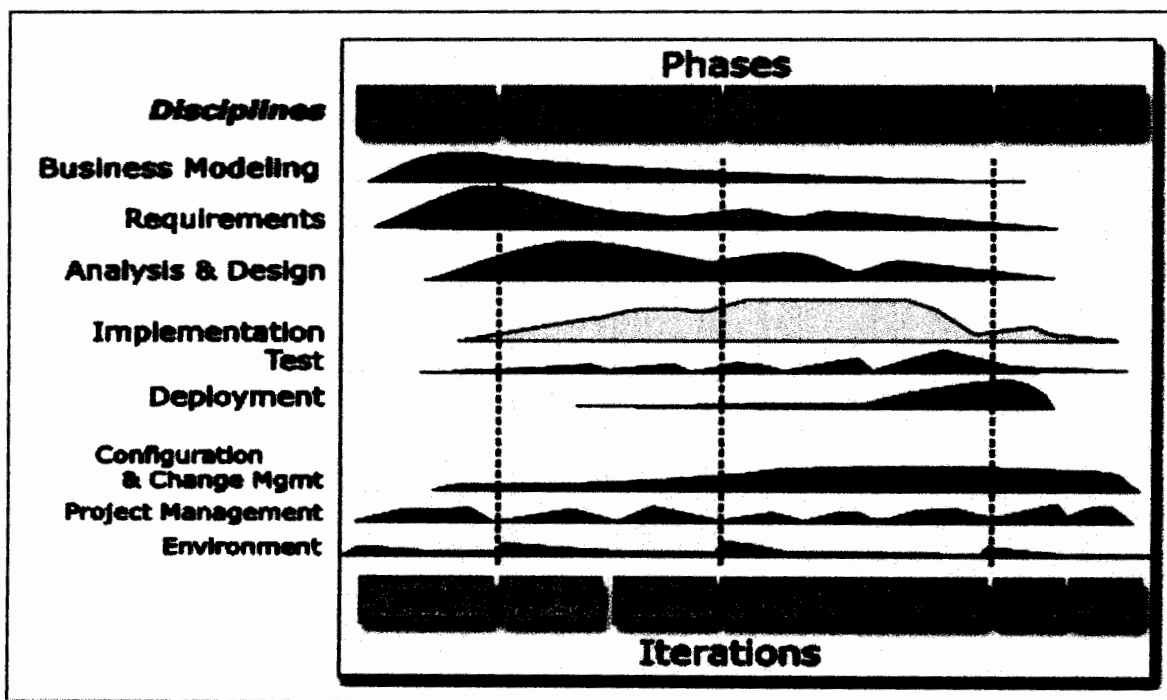
По време на итерация се изпълняват дейности, отнасящи се до идентифициране и анализиране на изискванията, имплементация, тестване.

Множество от тестови сценарии ще 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 набляга 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 подсили разработката на софтуер с високо качество, който удовлетворява нуждите на крайните потребители в рамките на предварително определен бюджет и срокове.



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.

Някои от практиките на RUP, които ще бъдат използвани при изпълнение на проекта, са:

Управление на изисквания

RUP описва ясен начин за извличане, организиране и документиране на необходимите функционални и нефункционални изисквания към системата и осигурява лесното графично дефиниране на бизнес процесите чрез средствата на UML. Изграждането на случаи на употреба (use cases) и тестови сценарии (test scenarios) е доказано като отличен начин за точно извличане на функционалните изисквания. Функционалността е движещ елемент в процеса на

проектиране, внедряване и тестване на софтуер. Ясното и точно дефиниране на изискванията е ключова предпоставка разработваната система да отговаря на крайните потребителски нужди.

Използване на компонентно базирана архитектура

Процесът се фокусира върху ранно развитие на архитектура и разделянето ѝ на компоненти. Това допринася за точна и коректна оценка на обхвата на проекта. RUP съдържа методологии за създаване на интуитивно разбираема и гъвкава архитектура, което подпомага компонентно-базираната софтуерна разработка. Компонентите представляват нетривиални модули или подсистеми, отговарящи на дадена функционалност. RUP предоставя систематичен подход за дефиниране на архитектура, разделяйки я на съставни компоненти и постигайки различни нива на абстракция.

Визуално представяне на софтуерната система

В резултат на този процес се изгражда графичен модел на системата, обхващащ както архитектурата и нейните елементи, така и взаимодействията между компонентите и цялостното поведение на системата. Визуалната абстракция предоставя възможност за изграждане на ясна представа за системата и обсъждане на различни нейни аспекти от различните екипи, без да се включва програмния код. Така се постига съгласуваност между дизайн и имплементация и се изгражда база за софтуерната реализация.

Итеративен подход

При разработката на днешните сложни софтуерни системи не е възможно на един етап да се определи целия проблем, да се проектира цялостно решение, да се разработи софтуер и накрая да се тества продукта. Итеративният подход позволява по-лесно да се прилагат тактически промени в изискванията, характеристиките или графика. Итеративното проектиране е подход на постоянно подобряване, прилагане, проверка и съгласуване на софтуерния продукт с изискванията на клиента. Итеративното проектиране позволява на Ръководителя на проекта да има по-голяма яснота за статуса на разработката, както и за обратната връзка с клиента.

Верифицира се качеството

Високата производителност, сигурност и надеждност са ключови фактори при разработването на съвременните информационни системи. RUP подпомага процеса на планиране, проектиране, изпълнение и оценка на тестването и осигурява контрол на качествените характеристики на разработвания продукт. Процесът по управление на качеството е паралелен на процеса на разработка.

Контролират се промените в софтуера

Управлението на промените включва идентифициране, мониторинг и контролиране на промените за успешно реализиране на софтуерното решение. Изолиране на процеса на разработка от промените е от огромно значение за успеха на проекта – промените могат да бъдат осъществявани само между две фази, а не по време на дадена фаза. При RUP се изисква ясна дефиниция на дейностите и целите в началото на фазата и анализ на резултатите в края на фазата.

Освен изброените практики в методологията RUP са заложили много от използваните от гъвкавите методологии практики:

- Сътрудничество с клиента – екипът по проекта и клиентът работят в условия на добра взаимна комуникация;
- Адаптивност – процесът позволява адекватно да се реагира на променящите се условия в динамична среда;
- Лекота – процесът е лесен за усвояване и модифициране.

Адаптирайки умело тези практики към утвърдения подход за софтуерна разработка на IBM превръща RUP методологията в силно средство за успех на проекта.

Управлението на софтуерната разработка обхваща следните дейности:

- Създаване на подробен план-график за изпълнение на дейностите по договора, който подлежи на одобряване от страна на Възложителя, в който ще се конкретизират дейностите и тяхната последователност, сроковете и обхватът на поръчката;
- Изпълнение на заложените в план-графика дейности;
- Обезпечаване с необходимите ресурси в съответствие с обхвата, обема и сроковете по съответната дейност;
- Ефективно разпределение на ресурсите по дейности;
- Ефективно координиране на всички дейности по време на изпълнение на поръчката;
- Непрекъснато взаимодействие между екипът на Изпълнителя и Възложителя през целия ход на изпълнение на проекта;
- Динамична преоценка и идентифициране на нови рискове;
- Управление на качеството.

Прилагането на добри практики при управлението на софтуерната разработка осигурява ефикасно планиране, проследяване и контрол на изпълнението на заложените дейности. Предпоставка за качественото изпълнение и управление на софтуерната разработка съгласно подхода RUP, са **добрите практики, характерни за този гъвкав подход:**

- Итеративно разработване на софтуерни решения;
- Управление на изискванията;
- Използване на компонентно-базирана архитектура;
- Моделиране, описание и визуализиране чрез UML;
- Проверка на качеството;
- Управление на промените.

Развойната среда за всички участници в гореописаната методология ще бъде Visual Studio Team Foundation Services 2012 (VSTFS). Управлението на проекта се осъществява чрез софтуерите за управление на проекти и ресурси MS Project и Visual Studio Team Foundation Server (VSTFS). Избраната системата VSTFS предоставя пълен набор от функционалности за всички елементи и фази от процеса:

- Приоритизиране на сътрудничеството между всички участници в процеса на разработване на системата и включване на клиента като равнопоставен член от екипа;
- Осигуряване на навременна обратна връзка с цел оптимизиране на вложените усилия и свързаните с тях разходи;

Предоставяне на лесни за ползване и подходящи инструменти за проектиране, разработка, тестване, внедряване и поддръжка на качествен софтуер.

Методология за софтуерна разработка

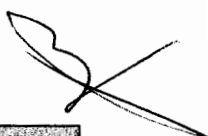
Съществуват множество методологии за обектно-ориентирана разработка на софтуер – от така наречените „леки“ и „динамични“ методологии (Agile Methodologies – <http://agilemanifesto.org/>) от типа на Scrum, RUP, Extreme Programming (XP), Dynamic Systems Development Method (DSDM) и др., до така нар. „тежки“ методологии като традиционния каскаден (waterfall) модел.

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.

За разлика от традиционния подход, при гъвкавият подход, какъвто е RUP, клиентът не чака до края на проекта, за да види резултата. Напротив - клиентът участва пряко в отделните етапи на проекта, а екипа от програмисти има чести разговори и срещи директно с клиента и неговия екип. 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.

За да покажем по-ясно разликите в двата подхода - **недостатъците на waterfall модела и преимуществата на RUP подхода, които допринасят за качествено изпълнение**, сме направили следната сравнителна таблица:


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 Изпълнителят 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,



Слаби страни на традиционния „waterfall“ технологичен подход	Преимущества на RUP подхода с оглед постигане на качествено изпълнение
	защото позволява няколко екипа да работят едновременно. Итеративността 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 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 Изпълнителя и Възложителя 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) Изпълнителят 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 Изпълнителят ще могат да контролират заедно развитието на проекта.

Изборът ни на RUP подхода 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



Изпълнителя 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.

Език за програмиране

За реализация на уеб базираното приложение за електронно съобщаване и призоваване екипът на „Смарт системс 2010“ ЕООД ще използва софтуерната платформа **Microsoft.NET Framework** и езика за програмиране **C#**. В .NET има много предварително написани класове. .NET е една от първите платформи, която идва с такъв богат набор от предварително написани класове. В .NET има и доста библиотеки, които могат да се добавят допълнително, което е изключително удобно при обектно-ориентираното програмиране.

Развойна среда

Развойната среда за всички участници в гореописаната методология ще бъде **Visual Studio Team Foundation Server 2013 (VSTFS)**. Избраната платформа предоставя пълен набор от функционалности за всички елементи и фази от процеса:

- Приоритизиране на сътрудничеството между всички участници в процеса на разработване на системата и включване на клиента като равнопоставен член от екипа;
- Осигуряване на навременна обратна връзка с цел оптимизиране на вложените усилия и свързаните с тях разходи;
- Предоставяне на лесни за ползване и подходящи инструменти за проектиране, разработка, тестване, внедряване и поддръжка на качествен софтуер.

Платформата предоставя пълна поддръжка за RUP процеса и всичките свързани с него роли и артефакти, без да се налага да се инсталира допълнителен софтуер при никой от участниците – представлява уеб базирана облачна услуга и е достъпна както през браузър, така и през софтуери като Microsoft Excel, Microsoft Project и много други.

Всички детайли за текущата итерация, както и задачите, на които е разбита и на кого са назначени, са достъпни през удобен интерфейс, откъдето с плъзгане (drag and drop) могат да бъдат приоритизирани и разпределяни между разработчиците на Изпълнителя.

Обратната връзка от клиента е от изключително значение за успешното разработване на качествен софтуерен продукт. Добрата комуникация и дефиниране на нуждите, които трябва да покрива системата, улесняват последващото и разработване без да се полагат излишни усилия. VSTFS поддържа специална добавка (Storyboarding Tool) за Microsoft PowerPoint, с която лесно могат да се проектират прототипи на потребителски интерфейс, които после да се използват от разработчиците и дизайнерите. По този начин още на етап проектиране изискванията към системата ще бъдат по-ясни и конкретни, което ще улесни последващото разработване.

В процесът на разработване и поэтапното публикуване на готовите модули, клиентът може да въвежда коментари чрез специално предоставен уеб базиран интерфейс (Feedback Manager), които да могат навременно да бъдат прегледани от мениджърите на проекта и разработчиците и при необходимост да се предприемат необходимите действия.

През Visual Studio разработчиците имат бърз и лесен достъп до всички задачи, които са назначени на тях, както и удобни инструменти за отразяване на прогреса по тях.

Всяка промяна в изходния код, която се вкарва в сървъра за управление на изходния код (Source Control Server), който се предоставя от VSTFS, се асоциира с дадена задача от текущия спринт. Така е много лесно и удобно да се проследи детайлно самият процес на разработване на елементите от системата, както и да се направи анализ за подобряване на планирането на следващия спринт.

VSTFS предоставя инструменти за следене на състоянието на системата дори след внедряването ѝ, за да могат да се предприемат навременно необходимите мерки при възникване на грешки и други непредвидени ситуации.

2.2. Управление на промените

Промени в обхвата на поръчката ще се правят само посредством стандартизирана процедура за управление на промени, като промени, свързани с предмета на поръчката, заложените цели и очакваните резултати са недопустими.

Под **промяна** обикновено се разбира разликата между изискванията, дефинирани в първоначалния договор между страните (често определени в самото тръжно задание) и изискванията, наложени след етапа на подписване на договора (обикновено по време на реалното изпълнение на проекта). Промените, които възникват при реализирането на проекта, могат да бъдат инициирани от Възложителя, Изпълнителя или дори трета страна по договора.

Промени може да се наложат поради изменения в:


- Обхвата на проекта;
- Първоначалните изисквания;
- Нормативната уредба;
- Ресурсите;
- Крайните срокове и др.

Всяка промяна в проекта оказва въздействие върху качеството и евентуално първоначално договорения срок за изпълнение. Целта на прилагането на процедура за управление на промените е контролираното и ефективно изпълнение на всички промени, както и минимизиране на влиянието, което различните искания за промяна могат да имат върху качеството на продукта/услугата, предмет на проекта, както и срока за изпълнение.

Съществуват задължителни процеси и техники, свързани с управлението на промените, които дават възможност на ръководителя на проекта да насочва и довежда проекта до успешен край, постигайки необходимото качество. Тези процеси и техники са в съответствие с действащите стандарти на Института за управление на проекти (PMI) и добри практики на ITIL.

В повечето проекти най-важният аспект на промяната е управлението на промените в обхвата.

"Обхват" е термин, използван за описание на границите на проекта. Обхватът се използва, за да се определи какво проектът ще създаде и какво няма да създаде. В повечето случаи на неуспех на проектите причината обикновено се дължи на два проблема – или екипът не е отделил достатъчно време за дефиниране на работата и/или е липсвало управление на



обхвата. Дори ако ръководителят на проекта е извършил добре дефинирането на обхвата, най-трудната част се състои в това да се управлява проекта в рамките на съгласувания обхват.


Целта на управлението на промените в обхвата е да запази приложимостта на одобрените бизнес изисквания и одобрения план за изпълнение.

Процедурата за управление на промените има за цел да осигури, че:

- Промените имат ясно определен и документиран обхват;
- Се одобряват само промени, които носят ползи на проекта;
- Промените са включени в плана за изпълнение на база приоритет и риск;
- Може да бъде демонстрирано по какъв начин промяната е:
 - породена, записана и класифицирана;
 - оценена относно влиянието, спешността, разходите, ползите и риска от промяната;
 - одобрена или отхвърлена от упълномощаващия промените, в зависимост от типа, обхвата и риска на промяната;
 - планирана, наблюдавана и докладвана;
 - изпълнена от посочения собственик на промяна;
 - изпитана, верифицирана и документирана като приключена от упълномощаващия промените.

Процесът за управление на промените преминава през следните стъпки:

- **Стъпка 1:** Член на екипа идентифицира необходимост от промяна на изискванията, след като планът на проекта е одобрен. Посредством унифицирана заявка за промяна се изпраща съобщение на ръководителите на проекта от страна на Изпълнителя и Възложителя. Заявката съдържа бизнес, техническа или управленска обосновка.
- **Стъпка 2:** Ръководителят на проекта анализира заявката и ако я одобри, се уведомява другата страна за преминаване към оценяване на приоритета ѝ, и се урежда среща за дискусия и анализ.
- **Стъпка 3:** Екипът за управление (определени от двете страни членове на екипите) анализира постъпилата заявка и въздействието ѝ върху проекта, след което я одобрява или отхвърля. Ако има отхвърляне се изпраща информация на инициращата страна и тя запазва историята за заявката в регистър на промените. При одобряване се оценява въздействието на промяната върху проектния обхват, срок и цена. Ако има такова въздействие, заявката се представя на надзорен съвет за одобрение (членове на надзорния съвет са например, представители на възложителя, изпълнителя, бенефициента и др. заинтересовани страни). Ако промяната няма въздействие върху обхвата, графика на изпълнение и цената се комуникира с инициращата страна чрез съответния ръководител на проекта. Одобрените заявки се отчитат в периодично в месечните отчети за напредъка по проекта.
- **Стъпка 4:** Ако заявката е одобрена и тя има въздействие върху обхвата, графика и цената се свиква надзорен съвет, който оценява заявката и прави предложение за приемането или отхвърлянето ѝ. Ако има отхвърляне се уведомява инициращата страна чрез представляващия я член на надзорния съвет. Запазва се история на заявката с нов статус в регистъра на промените. Ако заявката е одобрена, съответният



ръководител на проекта изпраща промяната на своя екип за реализиране. Одобрените заявки от надзорния съвет също се отчитат в периодично в междинните отчети за напредъка по проекта. Проектният график на изпълнение се актуализира в съответствие с одобрените заявки за промяната, а условията за обхвата, срока и цената за изпълнението на промяната се договарят официално (с Анекс или друг договор за възлагане).

Всички искания за промени и съответните резолюции ще бъдат документирани в „Регистър на промените“, който ще бъде актуализиран периодично в хода на изпълнение на проекта. След изпълнението на основни промени ще бъде проведен преглед, за да се провери, че:

- целите на промяната са постигнати;
- бенефициентите са доволни от резултатите;
- няма неочаквани странични ефекти.

Записите за промени периодично ще се анализират за установяване на повишени нива на промени, често повтарящи се видове, очертаващи се тенденции. Резултатите и заключенията от анализа на промените ще бъдат записани в Регистъра на промените и ще се предприемат необходимите действия.

При въвеждане на многобройни промени в хода на изпълнение на проекта, най-вероятно това означава, че:

- Изискванията и целите на Възложителя не са отразени или дефинирани добре по време на планирането на проекта;
- Липсват ясни обхват и изисквания на проекта;
- Липсва добра и ефективна комуникация между заинтересованите страни в рамките на проекта;
- Не са предвидени и избегнати някои рискове.

ЗАЯВКА ЗА ПРОМЯНА

Номер:		Дата:	
Инициатор:	Име	Организация	Контакти
Описание на промяната:			

Причина за промяната:				
Одобрение/отхвърляне на заявката:	<input type="checkbox"/>	Одобрена	<input type="checkbox"/>	Отхвърлена
		Ръководител проект (име)	Организация	Дата
	Коментар:			
Приоритет:	<input type="checkbox"/>	Критичен		
	<input type="checkbox"/>	Съществен		
	<input type="checkbox"/>	Среден		
	<input type="checkbox"/>	Добре е да го има		
Влияние върху обхвата:				
Влияние върху качеството:				
Влияние върху графика:				
Влияние върху ресурсите:				
Влияние върху рисковете:				
Влияние върху бюджета:				
Предложено решение:				
Одобрение/отхвърляне:	<input type="checkbox"/>	Одобрена	<input type="checkbox"/>	Отхвърлена
Коментар:				
Планирана дата за реализация:	Според плана на проекта (без промяна)			
За Възложителя:	Ръководител проект (име)	Организация	Дата	
За Изпълнителя:	Ръководител проект (име)	Организация	Дата	



Управление на промените ще се осъществява посредством **Team Foundation Server (TFS)** на Microsoft.

За управление на промените ще се използват **добри практики и процедури съгласно ITIL** (Information Technology Infrastructure Library) и по-специално ще бъдат изпълнявани подчертаните 2 процеса:

- Процес на управление на инцидентите;
- Процес на управление на проблемите;
- **Процес на управление на конфигурациите;**
- **Процес на управление на промените;**
- Процес на управление на връзките;
- Процес на управление на нивото на услугите;
- Процес на управление на капацитетите;
- Процес на управление на наличностите;
- Процес на управление на непрекъснатостта;
- Процес на управление на финансите;

Източници: <https://www.axelos.com/best-practice-solutions/itil>; <http://www.itil.org.uk/>.

2.3. Управление и отстраняване на установените несъответствия в софтуерната разработка

Предотвратяването, откриването и управлението на несъответствията е част от управлението на софтуерната конфигурация. Същността на тази процедура е след идентифицирането на дадено несъответствие/дефект да се предприемат мерки за неговото отстраняване и/или ограничаването/отстраняването му на следващ етап. Предотвратяването на появата на несъответствия се прилага като се изготви план за предотвратяване на несъответствия. Този план съдържа мерки за свеждане до минимум и отстраняване на недостатъци, индикатори за следене на несъответствията, както и анализ на причините за тяхното възникване.

Процедура за управление и отстраняване на установените несъответствия в софтуерната разработка преминава през следните основни стъпки:



1. Анализ на данните за несъответствие, провеждане на периодични прегледи и документиране на проведените тестове – резултатите се използват за приоритизиране и разделяне на несъответствията по първопричините за тяхното поява. Дефинират се метрики за несъответствията и се очертават проблемните области.



2. Идентифициране на стратегии за подобряване.



3. Създаване на ясна организационна йерархия за ескалиране на проблемите и комуникация.



4. Изготвя се план за действие. Чрез него се цели подобряване на процеса на разработка. Планът се преразглежда редовно и се проверява неговата ефективност. Ако се докаже, че заложените мерки са неефективни, планът бъде коригиран.



5. Извършване на периодични проверки дали планът за действие се спазва и отчитане на резултатите от неговото спазване.



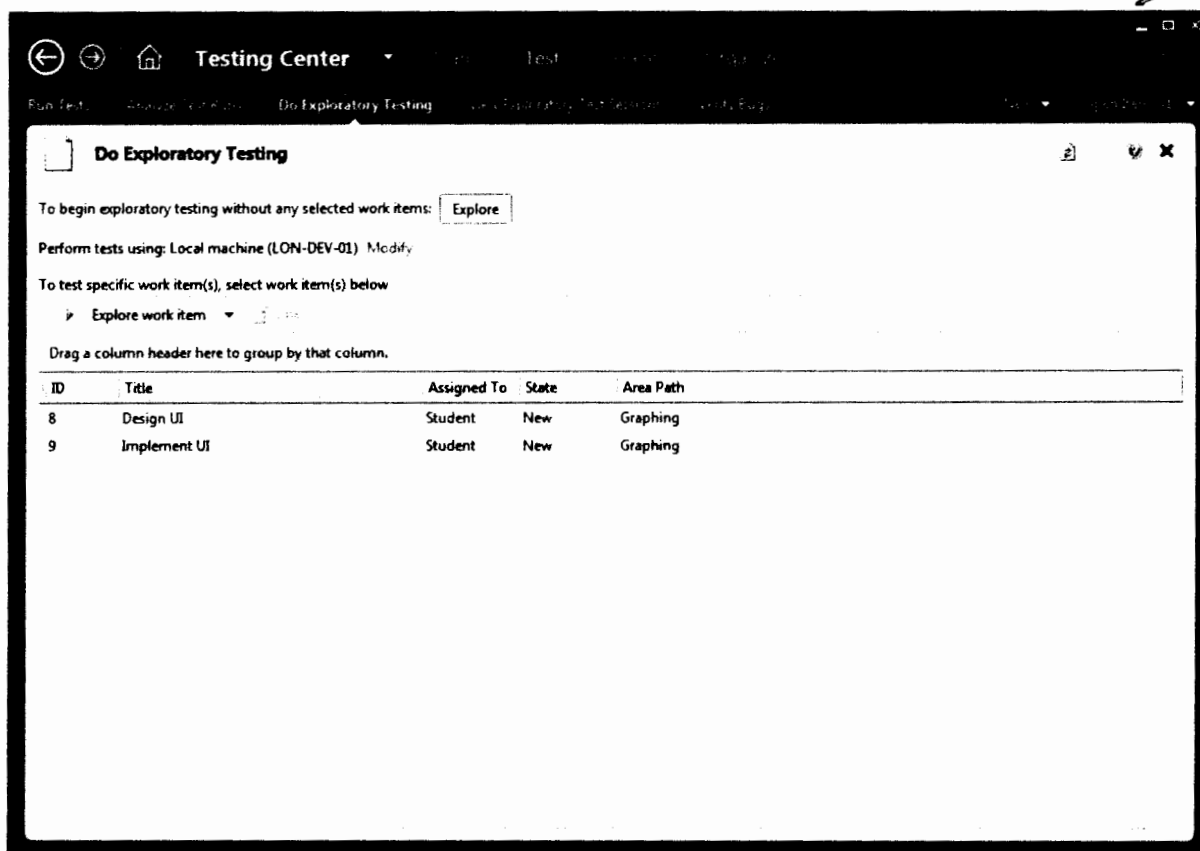
6. Редовен мониторинг и документиране на несъответствията, подредени по продължителност на съществуване и влияние.



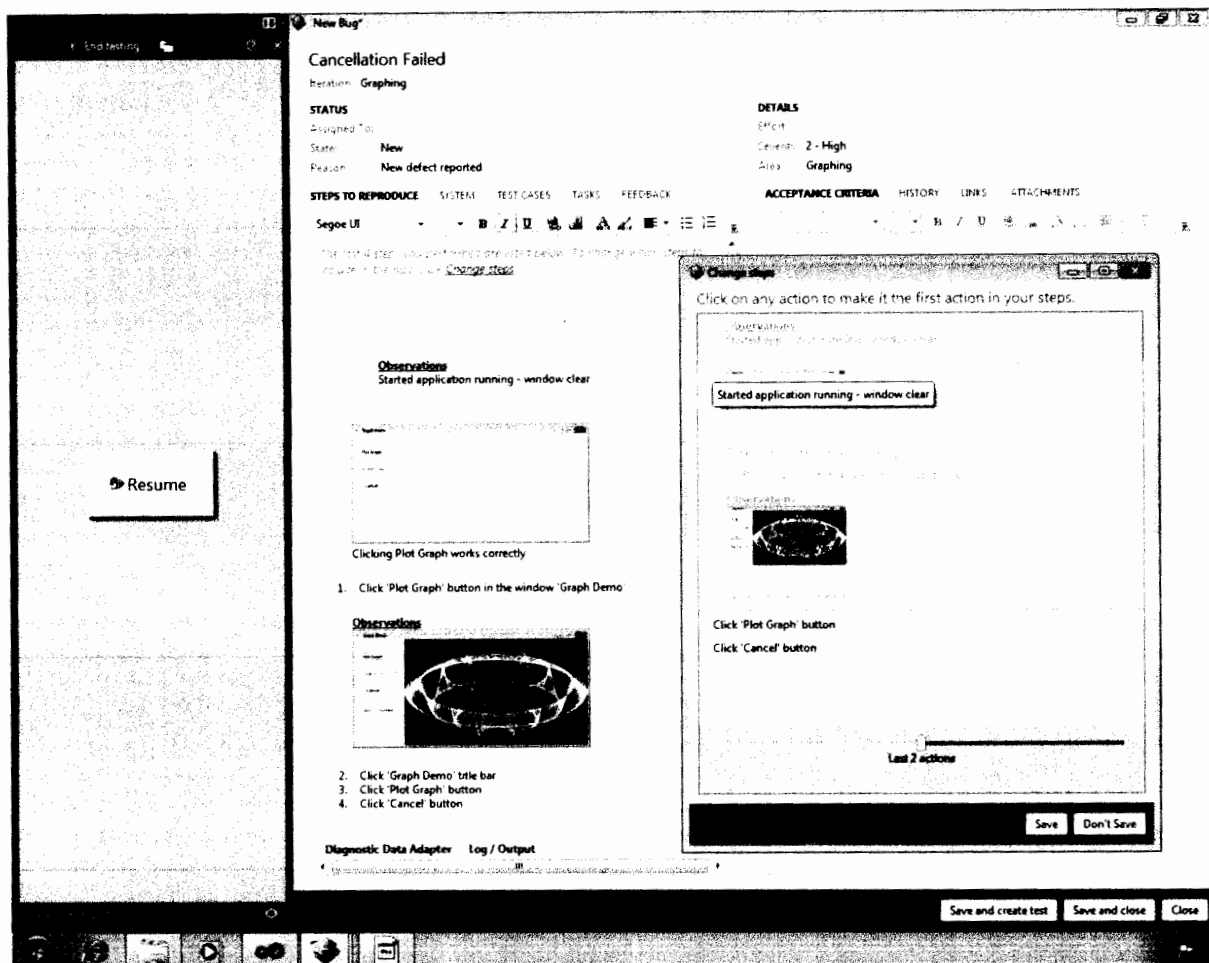
7. Приоритизиране и групиране на несъответствията в категории – критични, функционални, косметични (несъответствия, които засягат потребителския интерфейс и не влияят на коректното функциониране на софтуера).

Средата за управление на установените несъответствия в софтуера ще бъде VSTFS.

VSTFS предоставя пълна функционалност за дефиниране на тестови сценарии, както и за разработване и изпълнение на всички видове тестове, които ще бъдат заложили в текущия проект.




VSTFS предлага пълен набор от инструменти за проследяване и отстраняване на грешки (Bug Tracking System) и записва всички действия, извършени при проведените тестове, за да улесни идентифицирането на причините за възникналите грешки.



Управлението и отстраняването на установените несъответствия ще се осъществява посредством системата на Изпълнителя за проследяване на грешки и несъответствия, достъпна в Интернет. Системата позволява автоматични нотификации при регистриране, актуализиране, ескалация и решаване на инциденти.

Приоритетите на инцидентите, които се вписват в системата за сервизни заявки се определят в зависимост от влиянието им върху бизнес процеса/ите и работоспособността на приложението на ВСС. Редът за отстраняване на инцидентите се определя в зависимост от техния приоритет.


Приоритет	Въздействие върху бизнеса
!!!! 1 Критичен Immediate	Критично влияние върху бизнеса, изисква незабавно действие. В резултат от инцидента/повредата на технологично оборудване има пълно прекъсване на една или повече услуги, свързани с критични за дейността функции. Висок риск от финансови загуби и/или засягане на имиджа на ВСС и/ или потребителите на услугите на ВСС. Създава висок риск за компрометиране на информация в уебсайта. Нарушена комуникация с други страни/информационни



Приоритет	Въздействие върху бизнеса
	страни от критично значение.
III 2 Висок High/Urgent	Съществено влияние върху бизнеса. Влошаване на качеството на предлагана услуга или достъп до такава, без пълното ѝ прекъсване.
II 3 Среден Normal	Несъществено влияние върху бизнеса. Инцидентът има ограничено въздействие върху дейността, което засяга или създава неудобство за изпълнение на отделни функции, без да има цялостно отражение върху функционалността на уебсайта и работния процес на ВСС.
I 4 Нисък Low	Няма пряко влияние върху бизнеса в момента на възникването му. В момента липсва пряко влияние, но нерешаването му в определен срок крие потенциален риск от възникване на инцидент с по-висок приоритет.

Работният процес по обработка на заявките за грешки/несъответствия в системата е както следва:

1. Подаване на заявка за грешка/несъответствие. Възложителят може да подаде заявка по следните 3 канала:
 - a. Телефонен обаждаме, последвано от попълнена заявка в системата за проследяване на грешки и несъответствия;
 - b. Регистрирана заявка в системата на Изпълнителя с предоставено потребителско име. В този случай, Възложителят трябва да класифицира приоритетът на сервизната заявка (приоритетите са дадени в таблицата по-горе). Зададеният от Възложителя приоритет подлежи на промяна от страна на експерт на Изпълнителя, като промяната задължително се аргументира;
 - c. Електронно съобщение с придружена попълнена заявка за грешка/несъответствие в системата за проследяване на грешки и несъответствия.
2. Категоризиране на регистрирания проблем/грешка и определяне на приоритета.
3. Прехвърляне на задачата към експерт в зависимост от типа и приоритета на грешката. На базата на типа на грешката и приоритета ѝ, заявката се прехвърля (разпределя) към най-компетентния за случая експерт.
4. Диагностициране на грешка. Експерт на Изпълнителя извършва диагностициране на проблема с цел идентифициране на причината за възникването. Диагностицирането е отдалечено със съдействието на представител на Възложителя.
5. Отстраняване на проблема.
6. Изпълнена заявка. В този статус се достига след като са предприети действия и грешката е отстранена.
7. Проверка за изпълнена заявка в срок и качество на изпълнение. След изпълнението на заявката се проверява дали е изпълнена в срок и качеството на обслужване е на нужното ниво.
8. Ескалация към по-високо йерархично ниво в екипа на Изпълнителя и изготвяне на предложение за подобрене на процеса. В случай, че сервизната заявка не е изпълнена



в срок и/или качеството на сервизно обслужване не е на необходимото ниво, заявката се ескалира. Необходимо е да бъдат идентифицирани причините, поради които обслужването не е в срок или на ниво. След идентификация на тези причини се изготвя мотивирано предложение за подобрене с цел да се постигне по-високо качество на обслужване;

9. Затваряне на заявката. Действията по обработването на сервизната заявка приключват.

За управление и отстраняване на установите несъответствия ще се използват **добри практики и процедури съгласно ITIL** (Information Technology Infrastructure Library). ITIL е колекция от специализирани методологически принципи, синтезирани от най-добрите практики, с цел да се оптимизират процесите за ИТ услуги.

Най-известната част от ITIL са десетте базови процеса, обезпечаващи поддръжката и предоставянето на информационни продукти и услуги (IT Service Management – ITSM), като за управление на несъответствията ще бъдат изпълнявани подчертаните 4 процеса:

- **Процес на управление на инцидентите;**
- **Процес на управление на проблемите;**
- Процес на управление на конфигурациите;
- Процес на управление на промените;
- Процес на управление на връзките;
- **Процес на управление на нивото на услугите;**
- Процес на управление на капацитетите;
- Процес на управление на наличностите;
- **Процес на управление на непрекъснатостта;**
- Процес на управление на финансите;

Източници: <https://www.axelos.com/best-practice-solutions/itil>; <http://www.itil.org.uk/>.


2.4. Управление на качеството

Методика за управление на качеството

Целта на процесите по управление на качеството е да бъдат удовлетворени потребностите и очакванията, поради които е предприет проектът. Тези процеси включват всички дейности от цялостното управление на проекта, които определят политиката, целите и отговорностите по качеството и ги осъществяват чрез планиране на качеството, гарантиране на качеството, , качествен контрол и подобряване на качеството в рамките на системата за качество.

Планиране управлението на качеството

Планирането на управлението на качеството включва идентифициране на изискванията и стандартите за качество за конкретния проект и неговия продукт, както и начините за постигане на съответствие с тях. Това е един от ключовите процеси при планиране на качеството и се изпълнява паралелно с останалите процеси по планиране на проекта.



Системата по управление на качеството

В настоящия проект Изпълнителят ще използва стандартни процеси от своята Система за управление на качеството (СУК), която е сертифицирана за съответствие с международния стандарт ISO 9001:2008., включваща всички главни компоненти:

- Документация
- Ангажираност на ръководството.
- Внедряване на продукта.
- Мониторинг и измерване

За всяка дейност Отговорникът по управление на качеството ще разпределя ресурси и ще определя мерките за мониторинг на дейността, което ще намери отражение в изработването на Плана по управление на качеството.

Целите на системата по управление на качеството са:

- Да се проследят и открият причините за не съответстващ продукт и определят необходимите действия за предотвратяване повторно появяване на проблема;
- Анализ на всички процеси, операции, записи по качеството и мнения на клиентите, за да се открият и премахнат потенциалните причини за появяване на не съответстващ продукт;
- Превантивни действия за разрешаване на възникнали проблеми;
- Да създава списък с всички несъответствия, включващ крайна дата за отстраняване, с референция към документа за несъответствието и неговия статус;
- Да приложат контрол с цел установяване на ефективността на предприетите вече действия;
- Да се въведат промени в процедурите в резултат на извършените действия.

Анализът на всички несъответствия е задължение на отговорника по управление на качеството. Той оценява корективните действия от гледна точка на оперативни разходи, разходи за несъответствия, производителност и отношение на клиентите. Екипът от необходимите специалисти участва в корективните действия. След като бъдат предприети необходимите действия е необходим контрол, дали желаните цели са постигнати.

Съответните източници на информация трябва да бъдат идентифицирани. Примери за източници на информация са оплаквания на клиента, доклади за несъответствия, вътрешни доклади след проверка, резултати то преглед на управлението и резултати от собствена оценка.

Индикатори за наблюдение на развитието на проекта:

- Реализирана стойност. Използва се за преценка на графика и бюджета за оставащата част от проекта и/или за идентифициране на евентуални промени на обхвата.

Най-елементарният начин за измерване на реализираната стойност е чрез сумиране на първоначално планираните обеми работа за всички вече приключени задачи. "Процентът изпълнение" на проекта може да се изчисли, като реализираната стойност се раздели на общия обем първоначално планирани работи по проекта. Индикаторът за производителност е частното между реализираната стойност и реалния обем на работите, извършени за изпълнение на приключените задачи.

- Тенденции по отношение на дефектите. Използва се за прогнозиране на обема работа, необходима за отстраняване на дефектите.

Тенденциите по отношение на отворени и отстранени дефекти дават приблизителна представа за това, каква част от работата по отстраняване на дефектите е несвършена и в какъв срок дефектите биват отстранявани.

- Тенденция по отношение изпълнението на тестовите. Използва се за определяне на това, каква част от функционалността е реално завършена.

Най-меродавната мярка за завършеност е това, каква част от функционалността е внедрена. Ако всяка една от задачите по развоя представлява пакет от интегрирана функционалност, тогава би била достатъчна графика, показваща тенденцията на реализираната стойност.

Средства за управление на качеството:

За да изпълнят своите роли Ръководителят на проекта и отговорника по управление на качеството ще използват следните средства:

- Провеждане на вътрешен одит с цел да се направи преглед на управлението на процедурите, прилагане на методологията и други изисквания.
- Представяне на статуса от гледна точка на качеството пред отговорните лица по управлението на проекта.
- Прилагане на процедурите за корективни и превантивни дейности.
- Прилагане на подробен процесен подход за разрешаване оплакванията на клиента.
- Управление на измерване на качеството.
- Отговорникът по управление на качеството ще представи изискванията на потребителя и ще има критично влияние върху проекта.
- Отговорникът по управление на качеството ще бъде отговорен за подготовката и изпълнението на плана по качеството.
- Планът по качеството включва прегледи на изпълнението, проверка на документацията и анализ на риска.
- Ръководителят на проекта ще бъде отговорен за изпълнението на дейностите, определени в плана по качеството
- Отговорникът по управление на качеството ще следи и отчита всяко несъответствие с процедурите на системата за управление на качеството. Коригиращи действия ще бъдат предприети допълнително, ако е необходимо.

Осигуряване на качеството

Осигуряването на качеството включва всички планирани и систематични действия в рамките на системата за качество, които дават увереност, че проектът ще отговаря на съответните стандарти. Извърша се в хода на целия проект от вътрешни специалисти по качеството.

Процесите за осигуряване на качеството включват:

Контрол на качеството:

Дефинира се като набор от дейности, използвани за следене на разработката и проверка дали изискванията са удовлетворени. Използват се основно прегледи и отстраняване на дефектите преди завършване на системата. В рамките на проекта контролът на качеството

може да се осъществява както вътрешно в рамките на екипа, така и външно от нарочно звено по осигуряване на качеството.

За контрол на качеството ще се използват следните дейности:

- взаимни прегледи;
- одити;
- инспекции на програмния код и документацията;
- проверка на продуктите, които се доставят на Възложителя.

Прегледите ще бъдат извършвани в проектния екип на Изпълнителя, без протоколиране.

Управление на конфигурацията:

Целта на управлението на конфигурацията е да бъдат идентифицирани всички взаимосвързани компоненти на софтуера и документацията и промените в тях да бъдат контролирани по време на етапите от жизнения цикъл на системата.

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

Основните задачи на Управлението на конфигурациите са:

- Да специфицира версиите на съществуващите продукти и да предоставя информация за:
 - о Техния статус (например: в употреба, архивирани, готови за проверка на качеството)
 - о Кой притежава съответния продукт
 - о Връзките между продуктите
- Да поддържа актуални записи, съдържащи тази информация;
- Контролира промени в продуктите, като позволява тези промени да се правят само със съгласието на съответните власти;
- Одитира записите, за да потвърди, че те съдържат оторизираните продукти и само тези продукти.

Процесът по Управление на конфигурациите се състои от 5 основни функции:

- Планиране: какво ниво на Управление на конфигурациите ще се изисква за проекта и как това ниво да бъде постигнато.

- Идентификация: специфициране и идентифициране на всички компоненти на крайния продукт.
 - Контрол: след като веднъж един продукт е бил одобрен, мотото е „нищо не се премества, нищо не се променя без оторизация”.
 - Отчитане на статуса: записването и отчитането на всички данни, касаещи продукта.
 - Утвърждаване: серия от прегледи и одити с цел удостоверяване, че актуалното състояние на всички продукти, съответства на оторизирания статус в Управление На Конфигурациите записите.
- В този проект, задачата на Управлението на конфигурациите ще бъде да осигури:
- Механизмите за управление, проследяване и контрол върху всички продукти на проекта. Да съхранява файлове на всички продукти след преминаването им през проверка на качеството, да контролира достъпа до тях и да поддържа записи за техния статус.
 - Сигурно съхранение на всеки продукт по най-подходящия начин. Това включва контролиран достъп.
 - Възможност да събира различните компоненти, от които се състои окончателния продукт.
 - Система за регистриране, проследяване и съхранение на всички въпроси, свързани с проекта.

По този начин, Управление на конфигурациите позволява прецизен контрол върху продуктите на проекта.

За управление на конфигурацията Изпълнителят ще използва софтуерния инструмент Visual SVN Server. При необходимост, на упълномощени представители на Възложителя ще бъде предоставен достъп до системата за управление на конфигурацията.

Тестване

Тестването се използва да провери дали даден резултат съответства на поставените изисквания. Обхватът на тестването ще бъде определен в процеса на планиране на проекта, като най-общо тестването ще включва:

- тестване на компоненти;
- интеграционно тестване;
- системно/приемно тестване.

Дейностите по тестването включват:

- създаване на План за тестване;
- създаване на тестови случаи, сценарии, скриптове, пакети;
- зареждане на тестови данни;
- установяване на тестова среда;
- изпълнение на планираните тестове;
- управление на дефекти.

Тестването и приемането на конфигурациите ще включва следните етапи:

- Тестване на софтуера на място, преди внедряването в реална експлоатация. Конфигурацията на тестовия хардуер ще бъде идентична с тази на устройствата за реална работа, така че да подsigури инсталацията и конфигурацията на системата върху реалната работна среда с минимален брой стъпки;

- Цялостен тест, включващ осигуряването на всички придружаващи документи;
- Внедряване в реална експлоатация (в реална работна среда);
- Тестване на системата, включително осигуряването на цялата придружаваща документация.

Приемането на дейностите по разработка и внедряване се извършва по програма, изработена от Изпълнителя и утвърдена от Възложителя.

Приемните изпитания се считат за успешни, ако бъде установено безпроблемно функциониране на съответния модул, функционалност или модификация.

Ако по време на изпитанията бъде констатиран съществен или критичен проблем, то изпитанията се прекратяват и се провеждат отново след отстраняване на проблема.

За провеждането на приемните изпитания се съставят протоколи.

Тестването на системата в цялост и отделните ѝ компоненти ще бъде извършено в съответствие с подробен план за тестване, който следва да бъде съгласуван с Възложителя. В него ще бъдат конкретно разписани процедурите, тестовете, отговорностите и графикът за тестване.

Тестването ще бъде проведено на няколко нива, както следва:

- Тестване на компоненти. Има за цел да провери работата на отделните компоненти (модули). Конкретният обхват на компонентното тестване ще бъде уточнен след изготвяне на проекта на системата, където ще бъдат детайлно описани отделните компоненти с изискванията към тях. Тестването ще бъде проведено в информационната среда на разработката, без формално отчитане.

- Интеграционно тестване. Има за цел да провери комуникацията между отделните модули на системата, както и комуникацията на системата с други системи. Конкретният обхват на интеграционното тестване ще бъде определен след проектиране на архитектурата на системата, където ще бъдат открити отделните модули и връзките между тях, както и връзката с външни системи. Тестването ще бъде извършено в информационната среда на разработката, без формално отчитане.

- Системно тестване. Има за цел да провери доколко системата реализира заложените в функционални и други изисквания. Конкретният обхват на системното тестване ще бъде определен след утвърждаване на системния проект. Тестването ще бъде извършено в тестова среда на територията на Изпълнителя.

- Потребителско тестване, проверка за годност за експлоатация. Има за цел да провери годността на системата за експлоатация от потребителите в продукционна среда. Представлява пълно или частично повторение на системните тестове, но се провежда от предварително обучени потребители на Възложителя, в среда на Възложителя. Като допълнение могат да бъдат извършени и допълнителни тестове, например за пълно натоварване на системата за определяне на нейния пълен капацитет. Обхватът на потребителските тестове ще бъде съгласуван с Възложителя.

- Тестване при промени, регресионно тестване. Има за цел да провери дали след промени в системата не са възникнали дефекти в променяните или други нейни компоненти. Тестването се провежда след всяка промяна на интегрираната система. Тестовете, които следва да проверяват за регресия след промяна на отделните модули ще бъдат описани след изготвяне на проекта на системата. Ще бъде поддържана матрица на зависимостите, в която ще бъдат описани тестовете, които следва да бъдат извършени/повторени след промяна на всеки компонент.

Резултатите от тестовите ще бъдат представени в отчетен продукт от Фаза тестване, в който за всеки тест ще бъде предоставена следната информация:

- Справка за теста;
- Резултати;
- Описание на регистрираните нередности.

2.5. Управление на риска

Рискът е всяка потенциална трудност или последица, която има въздействие върху успеха на проекта или някакви странични ефекти. Правилното управление на риска повишава възможността за предвиждане, намалява разходите и времевата им рамка, повишавайки бизнес стойността, която носи.

Най-важната и първа стъпка за успешно управление на риска е да се определят рисковете и да се изгради техният списък.

Рисковете могат да бъдат категоризирани в следните категории:

- Рискове на бизнес процеса – рискове свързани с това колко добре познаваме бизнес процеса и какви са последиците за в бъдеще при неговата промяна;
- Рискове на изискванията – рискове, отнасящи се до това колко добре познаваме изискванията и последиците от тяхната промяна;
- Рискове на ресурсите – рискове свързани с наличността на ресурсите – персонал, финансови и други;
- Рискове, свързани с ограниченията, налагани от проекта;
- Рискове на развойната инфраструктура – рискове свързани с инструментите и инфраструктурата, използвана от екипа на разработчиците;
- Архитектурни – рискове свързани с нови архитектури, шаблони, библиотеки или технологии;
- Рискове на инфраструктурата – рискове, свързани с наличността, надеждността на инфраструктурата;
- Рискове, свързани с промените. Промяната на изискванията влияят силно върху успешния завършек на проекта.

Всички идентифицирани рискове ще влязат в Регистър с рисковете. Списъкът от рисковете влияе върху плана на проекта. При планирането трябва да бъдат избегнати повечето

от рисковете на бизнес процесите, изискванията, развойната инфраструктура, архитектурните рискове. За рисковете от другите типове трябва да има стратегия за минимизирането им.

Методология за управление на риска

Управлението на риска е систематичния процес по идентифициране, анализиране и реагиране на рисковете по проекта. То включва максимизиране на вероятността и ефекта от благоприятните събития и минимизиране на вероятността и ефекта от нежелателни за проекта събития.

Трябва да има ангажимент и от Възложителя, и от Изпълнителя за идентифицирането и контролирането на рисковете на проекта. Трябва да се осигури, че всички заинтересовани страни са информирани навреме относно вероятността за поява на рискове за проекта и относно възможните (планирани) мерки за реакция в случай на проявление – за елиминиране или минимизиране на неблагоприятния им ефект.

Изпълнителят ще създаде необходимата организация за изпълнение на всички процеси в цикъла на управление на рисковете:

Планиране управлението на риска – процес за определяне на подхода и дейностите по управление на риска. Важно е да се планират и последващите процеси по управление на риска, за да има съизмеримост между нивото, вида и прозрачността на управление на риска, от една страна, и самият риск и важността на проекта за организацията, от друга.

Идентифициране на риска – определяне на рисковете, които могат да повлияят на проекта, и документиране на техните характеристики в Регистъра на рисковете. Участници в процеса на определяне на риска са: основният екип на проекта, екипът по управление на риска, специалисти от други звена на фирмата, клиенти, крайни потребители, други ръководители на проекти и външни експерти. Определянето на риска е итеративен процес. Първата итерация може да се осъществи от част от екипа на проекта или от екипа по управление на риска. Целият екип на проекта и основните заинтересовани лица могат да осъществят втората итерация. Щом бъде идентифициран даден риск, се разработват и дори внедряват прости и ефективни мерки за преодоляването му.

Качествен анализ на риска – оценка на влиянието и вероятността от даден риск. Този процес приоритизира рисковете според евентуалното им влияние върху целите на проекта. Качественият анализ на риска е един от начините за определяне важността на дадени рискове и насочване на усилията към справяне с тях. Времето за реакция може да е критичен фактор при някои рискове. Оценката на качеството на наличната информация също спомага при преоценката на риска. Качественият анализ за оценка на вероятностите и ефекта на рисковете използва различни методи и средства.

Количествен анализ на риска - приложение на мощни статистически и други количествени методи за анализ на най-важните рискове за проекта. В този процес се използват методи като симулации "Монте Карло", дърво на решенията, анализ на чувствителността и други с цел получаване на количествени оценки за вероятността от проявлението на даден риск и на ефекта му върху срока, бюджета или характеристиките на продукта на проекта. Тези оценки са основа за избор на стратегия и за планиране на адекватни действия за реакция в случай на проявление на всеки риск.

Планиране реакция на риска - разработване на варианти и определяне на действия, които увеличават възможностите и намаляват заплахите за осъществяване на целите на проекта. Този процес включва възлагане на отговорности на отделни лица или звена във връзка с планираните дейности относно рисковете, както и използването на други необходими ресурси. Този процес гарантира адекватна реакция на идентифицираните рискове в случай на тяхното проявление.

Мониторинг и контрол - проследяване на идентифицираните рискове, наблюдаване на остатъчни рискове и откриване на нови рискове. Този процес служи за осъществяване на планираните действия за реакция на риска и за оценка на тяхната ефикасност. Той се изпълнява многократно в хода на проекта. С времето рисковете се променят, появяват се нови, някои очаквани рискове не се проявяват. Доброто наблюдение и контрол на рисковете дава информация, която подпомага взимането на адекватни решения за предотвратяване на неблагоприятните рискове и за използване на всички благоприятни фактори и условия.

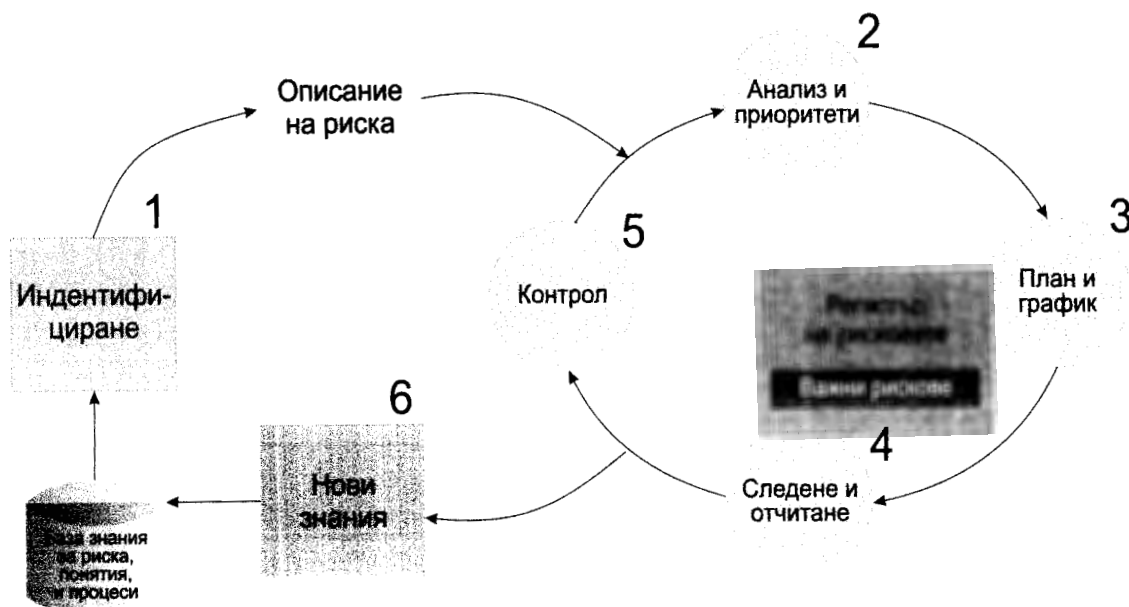
Контролът на риска може да включва избор на алтернативна стратегия, прибегване до резервен план, извършване на коригиращи действия или пре-планиране на проекта. Ръководителят на проекта и ръководителят на екипа за риска периодично получават информация за ефективността на плана и наличието на неочаквани влияния и взимат съответните мерки в хода на проекта.

Ключови фактори за успех

Управлението на риска може да бъде успешно при наличието на:

- Достъп до надеждна и актуална информация за рисковете
- Регулярна оценка и анализ на критичните рискове
- Непрекъснато следене и контролиране на ефекта от предприетите действия за реакция на рисковете
- Осигуряване на баланс между процесите за управление на риска и останалите процеси за управление на проекта

Процеси за управление на риска:



Идентифициране на рисковете

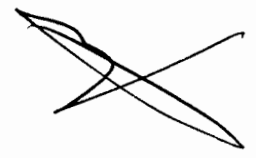
Тази стъпка идентифицира потенциалните рискове на проекта. Основни методи за идентифициране на рисковете в този проект ще бъдат:

- Периодична проверка и анализ на вътрешни и външни фактори, които имат пряка или косвена зависимост с резултати от проекта;
- Следене за възникване на събития, свързани с:
 - свързани проекти;
 - промени в законодателството;
 - отклонения от спецификациите;
 - предоставяне на информация необходима на продукт на проекта;
 - взимане на решения;
 - отделени ресурси и внимание от участниците в проекта;
 - промени в процедурите;
 - техническата среда;
 - сигурност на информация.

Веднъж идентифицирани, рисковете се документират в Регистъра на рисковете. Той съдържа детайли за всички рискове, тяхната оценка, собственици и статус в хода на проекта.

Оценка на рисковете

Оценката на рисковете се прави експертно въз основа на възможността да се случат и влиянието, което биха имали върху целите на проекта:

- 
- Възможността е оценената вероятност да се появи риска;
 - Влиянието е преценения ефект или резултат от появата на риска.

Влиянието се оценява на база на:

- Време;
- Разход;
- Качество;
- Обхват;
- Ползи;
- Хора/ресурси.

За оценка на вероятността за проявление на рисковете е възприета следната скала:

- Малка (М) – вероятност за проявление до 30 %
- Средна (С) - вероятност за проявление над 30 % до 70 %
- Висока (В) - вероятност за проявление над 70 %

За оценка на влиянието на рисковете е възприета следната скала:

- Незначително (Н)
- Слабо (Сл)
- Средно (Ср)
- Голямо (Г)
- Критично (К)

Стойностите за вероятност и оценка в предварителния регистър на рисковете са базирани на експертна оценка, основана на опита на Сمارт системс 2010 ЕООД в изпълнението на сходни проекти.

В различните случаи предложените действия за реакция на рисковете могат да се изпълняват едновременно или да бъдат разглеждани като алтернативи.

Избор на стратегия за реакция на рисковете

Изборът на действие е баланс между множество фактори. След идентифицирането и оценката на рисковете, е необходимо да се изготви и план за управление на риска, в които са описани контролните действия. Всяко контролно действие, от своя страна, е обвързано с асоцииран разход. Контролното действие е такова, че разходът за него трябва да е по-приемлив от риска, който контролира.

Основни стратегии/реакции срещу рисковете, които Изпълнителят използва при управление на риска, са:

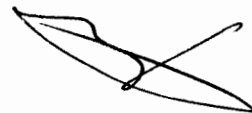
- **елиминация (Elimination)** – елиминиране на причината за идентифицирания риск;
- **избягване (Avoidance)** – предприемат се необходимите превантивни действия за премахване възможността за излагане на този риск (например: отлагане на тестове с цел да се подобри комуникационната среда);
- **превенция (Prevention)** – извършват се превантивни действия за намаляване на вероятността за реализацията на риска;
- **омекотяване/ ограничаване на ефекта (Mitigation)** – ако рискът не може да бъде избегнат, се извършват действия по намаляване на ефекта от евентуалното му осъществяване (например: подготовка на допълнителни ресурси, осигуряване на допълнително оборудване или средства);
- **планиране на решаването на проблеми (Contingency planning)** – изготвяне на план за отговор на потенциалния риск преди неговата поява и осъществяване;
- **приемане (Assumption)** – осъществяването на риска не може да се избегне. В такъв случай се осъществяват необходимите коригиращи действия за продължаване на планирания ход на проекта с минимални негативни резултати от въздействието на риска.

Изборът на стратегия за отговор на риск ще бъде базиран на задълбочен анализ на вероятността за осъществяване, обхвата на дадения риск, последиците върху проекта от евентуалното му осъществяване и оценка на разходите за реализиране на избраната стратегия.

Планиране и ресурсно обезпечение

Планирането включва:

- Определяне на количеството и типа ресурси, необходими за извършване на споменатите дейности;
- Разработване на подробен план за действие;
- Потвърждение на желанието за извършване на дейностите, идентифицирани по време на оценка на рисковете;
- Получаване на одобрение от ръководството;



- Определяне и възлагане на задачи на ресурси за извършване на определените дейности;
- Ресурсите, необходими за дейностите по превенция, редуциране и прехвърляне на рисковете, следва да се финансират от бюджета на проекта.

Мониторинг и отчитане

Изпълнителят ще обърне специално внимание на мониторинга и отчитането на дейностите по рисковете. Някои от дейностите ще включват наблюдение на идентифицираните рискове за промени в техния статус, а други ще включват:

- Проверка, че планираните дейности имат очаквания ефект;
- Наблюдение за ранни сигнали за поява на риск;
- Моделиране на насоки за предсказване на потенциални рискове;
- Проверка, че цялостното управление на риска се прилага ефективно.

План за изпълнение на проекта

за изпълнение на обществена поръчка по реда на Глава осма „а” от ЗОП с предмет „Оптимизация на бързодействието на Автоматизираната система за управление на делата АСУД и интегриране с единната среда за безхартиен обмен на дела във Върховния касационен съд”

1. Въведение

Настоящият документ представя нашето виждане за изпълнението на проекта, като се дефинират целите, обхвата, организацията на проекта, графика за изпълнение и екипа за изпълнение. В него са описани отделните дейности в проекта, очакваните резултати, процесите за управление и свързаните с тях планове.

2. Цели на проекта

Основна цел на проекта е оптимизация на бързодействието на Автоматизираната система за управление на делата АСУД и интегриране с единната среда за безхартиен обмен на дела във Върховния касационен съд.

Подцели на проекта

- Трансформация на модела на данни на АСУД на ВКС с цел оптимизация на бързодействието, чрез обособяване на отделни бази данни за управление на информацията на основни групи информационни обекти – база данни „Дела”, база данни „Страни”, база данни „Заседания”, база данни „Съдебни актове”.
- Извършване на миграция на данните от съществуващата база данни на АСУД на ВКС към трансформирания модел от отделени бази данни за информационни обекти.
- Интеграция на АСУД на ВКС с единната среда за безхартиен обмен на съдебни дела, изградена от ВСС, позволяваща получаване на електронни папки на изпратени за разглеждане дела. Изпращане на данните за делото, заедно с резултата от разглеждането му във ВКС в електронен вид към съдилищата, след взимане на решението.
- Добавяне на нови полета в базата и работните форми с оглед промените в НПК от 2015г.

3. Обхват на проекта

По време на изпълнение на проекта ще бъдат извършени следните основни дейности:

- Анализ на модела на данни и изготвяне на предложение за оптимизация на модела на данни.
- Анализ на стандарта за безхартиен обмен на съдебни дела и изготвяне на предложение за интеграция на АСУД на ВКС.
- Изграждане на прототип на актуализираната система с новоразработения модел на данните.
- Изграждане на прототип на механизма за обмен на дела.
- Разработка на миграционен механизъм за разпределяне на данните в обособените бази данни по видове документи.
- Тестване на миграционния механизъм чрез извършване на миграция на данните към прототипа.
- Тестване на механизма за обмен на дела чрез извършване на обмен на данни от прототипа.
- Извършване на миграцията в продуктивна среда.
- Внедряване на актуализираната версия на АСУД на ВКС

4. Етапи на изграждане

Жизненият цикъл на софтуерната разработка при подхода RUP се състои от 4 основни последователни етапа:

- Бизнес и системен анализ (Inception);
- Проектиране (Elaboration);
- Разработка (Construction);
- Внедряване (Transition).

Етапите на изграждане са:

Анализ и проектиране, изграждане на прототип на системата

Етапът Бизнес и системен анализ включва стартирането на проекта и провеждане на аналитични дейности с цел ясно дефиниране на целите, обхвата и изискванията

Проектните резултати от тази дейност предоставят решения за определяне на проблемната област за събиране и следене на промените в изискванията, моделиране на взаимодействията и определяне архитектурата на системата. Моделирането на процесите и

взаимодействията между компонентите и архитектурните предложения са основния начин за връзка с Възложителя и задоволяване на изискванията му в най-голяма степен.

При RUP провеждането на анализа на бизнес процесите и описанието на изискванията е покрито от две дисциплини:

- Бизнес моделиране (Business Modeling Discipline);
- Специфициране на изискванията (Requirements Discipline);

Целта на бизнес моделирането е да бъдат разбрани бизнес процесите в организацията, имащи отношение към разработваната софтуерна система. Планирането на промени в работата, често обвързано с внедряване на софтуерни системи, дори когато са минимални, изисква внимателен анализ на настоящата ситуация и на изискванията към новия процес и системата, която ще го обслужва. Единствено качествено изпълненият бизнес анализ дава сигурността, че една софтуерна система ще бъде изготвена в съответствие с нуждите на Възложителя и че ще бъде внедрена без проблеми за ползване в неговата организация.

Бизнес процесите описват извършваните дейности от потребителите на високо ниво, заложили в съответните правила и инструкции за работа, абстрахирайки се от възможните системи, които могат да ги подпомагат. Работейки заедно с Възложителя, бизнес аналитикът на Изпълнителя ще извърши следните дейности, които в следващия етап ще бъдат документирани в детайлната спецификация на софтуерните изисквания:

- Детайлно описание на системата (бизнес процесите; ролята и отговорностите на потребителите);
- Описание на случаите на употреба (use cases).

Разбирането и оптимизирането на системата още на ниво бизнес дейности и процеси в последствие ще доведе до реализиране на софтуерна система, максимално отговаряща на целите на проекта и близка до нуждите на Възложителя. В резултат на тези дейности ще се определи концепция за разработване на информационната системата, която да реализира техническата спецификация.

Архитектурни решения ще бъдат изразени чрез няколко гледни точки/изгледа:

- **Логически изглед** – показва декомпозицията на решението в отделни логически елементи (подсистеми, компоненти, класове и т.н.);
- **Изглед на процесите** – показва взаимодействието между елементите на логическия изглед и същинските софтуерни процеси и нишки на приложението;
- **Физически изглед** – показва връзката между процесите и физическия хардуер, върху който се изпълняват;
- **Изглед на данните** – описание на системата от гледна точка на съхранението на данните и съответния носител.

Резултатите от този етап са входна информация за етап „Проектиране“.

Проектиране (Elaboration)

Етап Проектиране на системата включва детайлизиране и документиране на технически решения и извлечените изисквания към системата от предишния етап. Много важно е през този етап да се стабилизира в достатъчна степен изискванията и плановете, за да могат върху тях да стъпят основните дейности по дизайн и реализация на софтуерното решение.

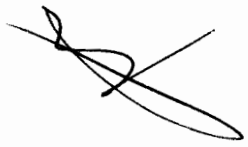
Проектните резултати от този етап включват модела на архитектурата, дизайна на системата, модулите на системата на различни нива, взаимовръзките между отделните компоненти, модела на базата данните, функционални и нефункционални изисквания и стратегия за нейното конструиране.

По време на този етап Изпълнителят ще изготви детайлна спецификация на софтуерните изисквания, в която ще бъдат документирани:

- Функционалните изисквания към софтуерното приложение;
- Нефункционалните изисквания към софтуерното приложение;
- Системните изисквания – описания на функциите, услугите и работните ограничения на софтуера;
- Дефинирани роли на потребителите.

Детайлната спецификация на софтуерните изисквания ще включва всички сценарии на употреба (use cases), които описват взаимодействието на потребителите със системата.

Отделните софтуерни компоненти, 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. Проблемният домейн се декомпозира на отделни обекти, следвайки принципите на дизайн от общото към частното и свързване на частите съобразно техните отговорности. Обектно-ориентираният 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, които са доказали своята ефективност при много различни ситуации и се препоръчват от водещите производители на софтуерни решения. Такъв шаблон за дизайн представлява декомпозицията на софтуерното решение на отделни слоеве, комуникиращи помежду си по строго определени интерфейси. Основно предимство при тази архитектура 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. Всеки слой в последствие се декомпозира на



отделни модули съобразно обектно-ориентирания подход. При централизираните системи доказан подход е разделянето на следните слоеве:

- Слой на базата данни;
- Слой на бизнес логиката;
- Слой на потребителския интерфейс
- Слой на интерфейса с други системи,

Разработка (Construction) и Тестова експлоатация на прототипа в продуктивна среда

Разработката обхваща цялостната реализация на софтуера и неговото тестване. През този етап екипът на Изпълнителя ще извърши програмиране на софтуерните модули, съгласно одобрената детайлна спецификация на софтуерните изисквания.

Основните дейности включени в този етап са:



- Писане на програмен код;
- Създаване на компонентите и модулите на уеб базираното приложение;
- Вътрешно тестване;
- Изготвяне на тестови сценарии на разработените компонентите и модулите на системата (План за приемателно тестване);
- Изготвяне на техническата и експлоатационна документация на проекта;
- Изготвяне на план за внедряване на новите софтуерни модули.

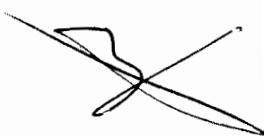
Внедряване (Transition)

Главната цел на етап Внедряване е успешното внедряване на разработеното софтуерно решение и гарантиране, че то отговаря на нуждите и изискванията на потребителите. Системата се оценява и усъвършенства на базата на обратна връзка от потребителите.

Внедряването на системата се извършва след приключване на етап Разработка на системата (Construction). Този етап включва инсталиране, настройка и тестване, миграция на данни (ако е необходимо), установяване на връзки с комуникационната среда и други информационни системи, дефинирани в обхвата на проекта, комплектуване на пълна документация, която съдържа информация за функционирането (техническа гледна точка) и използването на системата (гледна точка на потребителите). Етапът приключва с предаване на напълно функционираща версия на системата, включително предаване на техническа и експлоатационна документация.

В края на етапа системата се пуска в реална експлоатация и се извършва оценка на цялостното изпълнение на проекта.





Системата ще бъде инсталирана на определено от Възложителя място/места и в съгласувано с Възложителя време.

След инсталация софтуерните специалисти на Изпълнителя ще направят необходимите верификации за коректност. Ще се извършат тестове и наблюдения, които гарантират, че по време на работата на системата базата данни остава консистентна, запазва своята референциална цялост, няма липсващи, дублирани записи или такива, съдържащи невалидна стойност и т.н.

Тестовите за приемане на системата (приемателните тестове) ще се проведат след като системата е инсталирана в реална работна среда. При необходимост ще бъдат предприети коригиращи мерки с цел пълно удовлетворяване на изискванията към софтуерния продукт.

Основни дейности през етапа:



- Подготвяне на средата за инсталиране и експлоатация на разработения софтуер;
- Инсталиране в реална експлоатационна среда и настройки
- Интеграция с външни системи;
- Провеждане на приемателни тестове;
- Постигане на съгласие със заинтересованите страни, че разработената система отговаря на цялостната визия на проекта и дефинираните критерии за успех;
- Финализиране на техническата и експлоатационната документация;
- Пускане на системата в реална експлоатация.

Срок за изпълнение – 3 (три) месеца, считано от датата на подписване на договор.

5. Организация на Изпълнителя

Разработката на софтуер е 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. Не случайно един от създателите на RUP Grady Boch казва „Време беше да си призная и да разкрия една от мръсните малки тайни на софтуерното инженерство: Хората са по - важни от всеки процес“. Ето защо, създаването на правилна организация за изпълнение на проекта и точното дефиниране и разпределение на ролите на участниците в екипа 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 вътрешна (при изпълнителя) организация за изпълнение на проекта и разпределение на ролите на участниците в екипа, базирана на принципите и най-добрите практики на RUP за прозрачност, приемственост и възможност за обмен на ноу-хау, както в рамките на екипа за изпълнение на проекта, така и от и към екипите, участващи в изпълнение на проекта от страна на Върховния касационен съд.



Съгласно изискванията, заложи в Техническата спецификация в екипа на Изпълнителя се предвижда екипа да бъде съставен от ръководител на екип и експерти.

Ръководител на екип

Методологията RUP дефинира обхвата от отговорностите на Ръководителя на екипа за изпълнение на проекта в следния ред – хора, продукт, процес, проект. Ръководителят на екипа е отговорен за поддържането на плана през целия жизнен цикъл на проекта и за представянето му в края на всяка фаза с детайлна информация за следващата фаза и при необходимост, актуализация на информацията за остатъка от проекта.

По-конкретно, Ръководителят на екипа ще отговаря за оперативното управление на проекта, което включва пълния набор от задачи и отговорности през целия жизнен цикъл от изпълнението на проекта – разпределение на работата между членовете на екипа за изпълнение на проекта, следене за срочното и качествено изпълнение на дейностите и предаване на отчетните резултати, осъществяване на ежедневната комуникация с клиента, а така също и осигуряване на координация между управленското и изпълнителското ниво.

За ръководител на екип от страна „Смарт системс 2010“ ЕООД предлагаме **г-н Марин Кошутков**, притежаващ богат опит в изпълнението на ИТ проекти със сходен обхват и сложност, най-вече в съдебната сфера и притежаващ валиден сертификат за владение на технологията, използвана при разработка на проекта – IBM Lotus Notes/Domino, завършено висше образование в областта на информатиката и богат опит в разработката на приложения в областта на настоящата поръчка.

Експерт

Съгласно Техническата спецификация, предлагаме оперативното ниво на екипа за изпълнение на проекта да се състои от 1 експерт с опит в разработката на приложения в областта на настоящата поръчка и притежаващ валиден сертификат IBM Lotus Notes/Domino.

Реализацията на един ИТ проект е невъзможна без на пръв поглед рутинната, но всъщност изключително важна роля на програмистите, които носят отговорността да разработят информационната система в пълно съответствие както с функционалните, така и с нефункционалните изисквания на системата, напътствани при изпълнение на техните задачи от ръководителя на екип.

Отчитайки критичността на ролята на програмистите в проекта, ние предлагаме като ключов експерт **г-н Димитър Миладинов**, който има дългогодишен и задълбочен опит, както

в технологиите на които се базират системите във ВКС, така и опит в областта на информатиката и богат опит в разработката на приложения в областта на настоящата поръчка.

Г-н Димитър Миладинов притежава валиден сертификат за владение на IBM Lotus Notes/Domino и завършено висше образование.

Специалист по качеството/Отговорник по управление на риска

Тази роля е изключително важна за качествено и успешно изпълнение на проекта. Двете дейности взаимно се допълват, за да гарантират, че всички изисквания на ВКС ще бъдат качествено изпълнени и предадени в срок. Това налага много високи изисквания и отговорности на специалиста в тази роля. В тази връзка, ние считаме за най-подходяща и предлагаме тази роля да бъде заета от **г-жа Лора Петрова**, заради нейният богат опит в изпълнението и поддръжката на ИТ проекти, следвайки най-добрите практики на софтуерното инженерство и спазвайки международните стандарти за качество на ИТ услугите и продуктите. Тя ще е отговорна и за цялостния процес по Управление на риска и прилагането на предложената Методология за управление на риска.

6. Гаранционна поддръжка

Изпълнителят ще гарантира, че разработената софтуерна система няма дефекти, произтичащи от дизайна, реализацията, внедряването или други елементи на процеса за разработка.

Гаранционната поддръжка обхваща всички действия, които имат за цел да запазят работоспособността на системата като цяло или на някой неин компонент или да възстановят системата в работоспособно състояние (в съответствие с техническото задание).

Определението „работоспособност“ обхваща и отстраняването на грешки и несъответствия (бъгове), които може да са на лице по време на инсталация на системата, но да не са били открити по време на приемателните тестове, както и адаптиране на системата към променящата се среда на работа (или промени в нормативната уредба).

Целта на гаранционната поддръжка е да осигури пълно функциониране и оптимална работоспособност на системата по време на гаранционния ѝ период. Успешното изпълнение на гаранционната поддръжка включва:

- Наблюдение на работоспособността на системата;
- Отстраняване на грешки във функционирането на системата.
- Актуализация на функционалността на системата.

„Смарт системс 2010“ ЕООД ще осигури гаранционна поддръжка за период от 1 (една) година. Гаранционният период започва от датата на подписване на протокол за приемане на системата в редовна експлоатация.

Наличието на поддръжка за времето на гаранционния период на системата ще бъде гарантирано като за целта ще бъде определен човек от екипа (single point of contact), който ще отговаря във всеки един момент за поддръжката. По време на гаранционния период ще бъде използвана уеб базирана Help Desk система за регистриране и проследяване на дефектите, която да се ползва съвместно от представители на Изпълнителя и Възложителя.

Изпълнителят се задължава на всеки три месеца в рамките на гаранционния срок да извършва профилактика и проверка за нормалното функциониране на системата. По време на гаранционното поддържане на информационната система Изпълнителят следва да предоставя актуализирана документация, ако отстранените грешки или софтуерни промени налагат това.

При изтичане на гаранционния срок се предава актуализирана версия на програмния продукт и свързаните с него продукти, в които се отразени всички корективни дейности, извършени от Изпълнителя.

За срока на поддръжката Изпълнителя се задължава да предостави ниво на достъпност на предоставяните от системата услуги /SLA/, с параметри в съответствие с техническото му предложение.

Не се предвиждат допълнителни плащания за дейността през периода на гаранционна поддръжка. Сумите за тези дейности трябва да са включени в договорната сума.

Изпълнителят ще осигури гаранционно обслужване на системата, която включва:

- Извършване на диагностика на регистриран проблем с цел осигуряване на правилното функциониране на системите;
- Отстраняване на открити грешки в системата и функционални откази;
- Съдействие при промяна в настройките на системата;
- Консултации за разрешаване на проблеми по предложената конфигурация на средата (операционна среда, база данни, middleware, хардуер и мрежи), използвана от системата;
- Консултации за разрешаване на проблеми свързани с промени в конфигурацията на софтуерната инфраструктура на мястото на инсталацията;
- Възстановяване на системата и данните при евентуален срив на системата. В този случай Изпълнителят ще осигури екип по поддръжка при Възложителя до крайното възстановяване на системата;
- Експертна помощ на потребителите на софтуера по телефон и електронна поща в рамките на работното време (от 9:00 до 17:30 часа всеки работен ден от седмицата).

- Актуализация на документацията на системата в резултат извършени действия/промени в рамките на поддръжката и предаване на Възложителя.
- Осъществява промени по функционалността на информационната система, произтичащи от технологични или нормативни изисквания и след заявяването им от страна на Възложителя
- Предоставяне на уеб базирана система (bug-tracking system) за регистриране и проследяване на дефекти (бъгове), която ще се използва съвместно от представители на Възложителя и Изпълнителя;
- Дейности по осигуряване на експлоатационната годност на софтуера и ефективното му използване от Възложителя;
- Подобряване на сигурността на системата:
 - Инсталация и пускане в експлоатация на всички подобрения и допълнения в текущата версия на системния софтуер;
 - Актуализиране на настройките на системния софтуер и операционните системи при разкриване на уязвимости;
- промени по функционалността на информационната система, произтичащи от технологични или нормативни изисквания и след заявяването им от страна на Възложителя

7. Отчетни документи

Изпълнителят ще предостави следните отчетни документи по време на изпълнение на проекта.

- Общ план на проекта - подробен план на етапите на изпълнение на проекта;
- Детайлната спецификация на изискванията;
- План за приемателно тестване;
- План за гаранционна поддръжка.

8. Гант диаграма

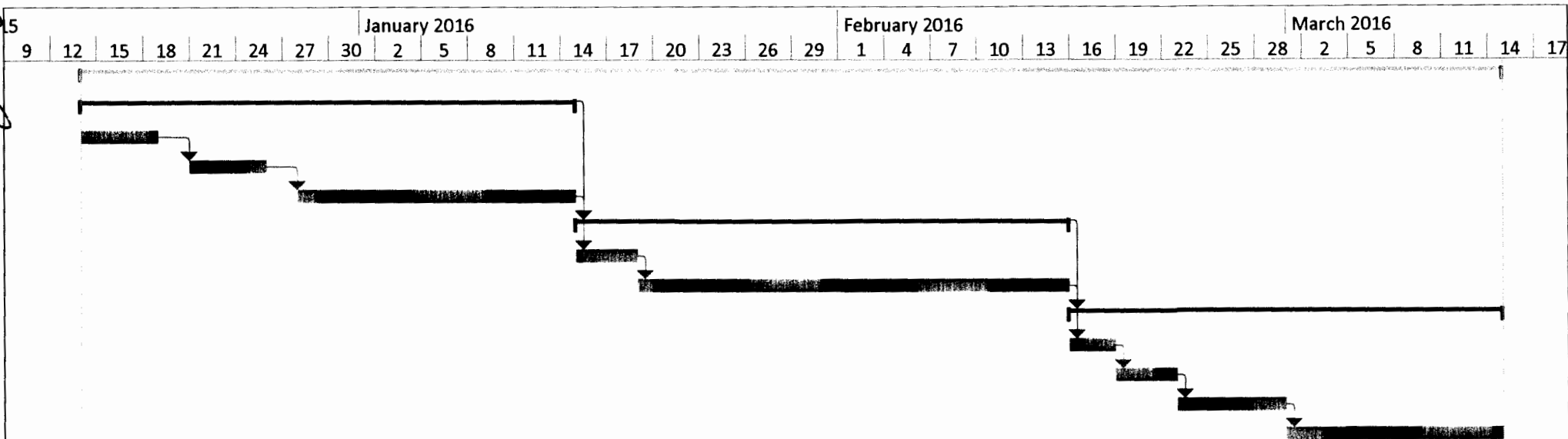
В Приложение № 1 е предложен график за изпълнение на обществената поръчка в Gantt Chart формат. В графика са описани детайлно дейностите, като е показана тяхната взаимовръзка и последователност, както и времето за предаване на всички функционалности, документи и други проектни резултати, предвидени за изработване в обхвата на обществената поръчка.

Handwritten: 12 Dec

ID	Task Name	Duration	Start	Finish	er 2015
0	Разработване и внедряване на единен портал на електронното правосъдие, и обучение за работа с	66 days	Mon 14.12.15	Mon 14.3.16	6
1	ЕТАП 1 - АНАЛИЗ И ПРОЕКТИРАНЕ, ИЗГРАЖДАНЕ НА ПРОТОТИП	24 days	Mon 14.12.15	Thu 14.1.16	
2	Анализ на модела на данните	5 days	Mon 14.12.15	Fri 18.12.15	
3	Проектиране на промените в системата	5 days	Mon 21.12.15	Fri 25.12.15	
4	Изграждане на прототип	14 days	Mon 28.12.15	Thu 14.1.16	
5	ЕТАП 2 - ТЕСТОВА ЕКСПЛУАТАЦИЯ	22 days	Fri 15.1.16	Mon 15.2.16	
6	Тестова миграция	2 days	Fri 15.1.16	Mon 18.1.16	
7	Тестване	20 days	Tue 19.1.16	Mon 15.2.16	
8	ЕТАП 3 - ВНЕДРЯВАНЕ	20 days	Tue 16.2.16	Mon 14.3.16	
9	Инсталиране и настройка на софтуера (софтуерните модули) в реална работна среда на място при	3 days	Tue 16.2.16	Thu 18.2.16	
10	Окончателна миграция	2 days	Fri 19.2.16	Mon 22.2.16	
11	Провеждане на приемателни тестове	5 days	Tue 23.2.16	Mon 29.2.16	
12	Пускане на софтуера в редовна експлоатация	10 days	Tue 1.3.16	Mon 14.3.16	

Project: Разработване и внедряване Date: Mon 7.12.15	Task		Inactive Task	Start-only	
	Split		Inactive Milestone	Finish-only	
	Milestone		Inactive Summary	Deadline	
	Summary		Manual Task	Critical	
	Project Summary		Duration-only	Critical Split	
	External Tasks		Manual Summary Rollup	Progress	
	External Milestone		Manual Summary		

Handwritten signature/initials

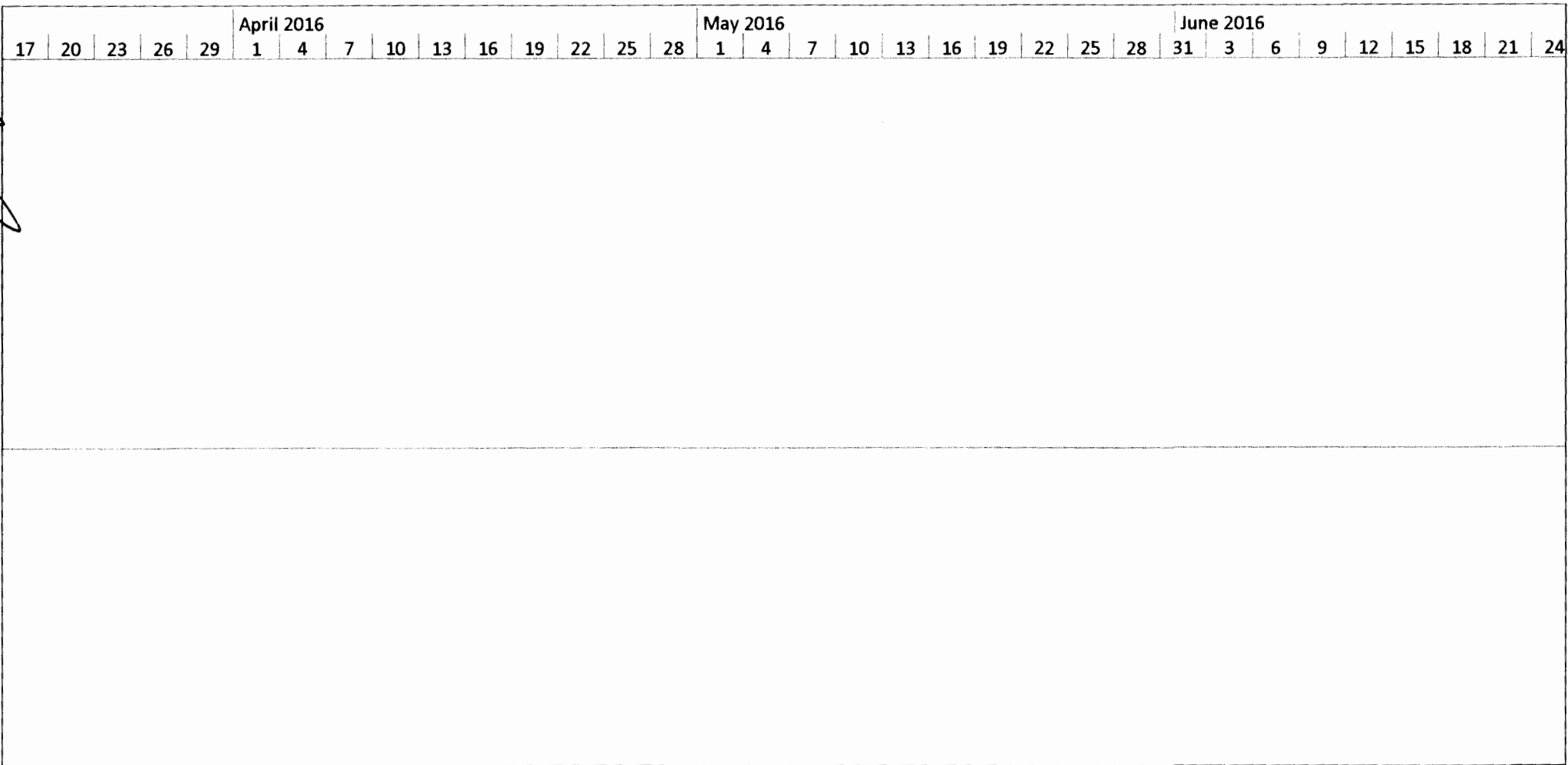


Project: Разработване и внедряване Date: Mon 7.12.15	Task		Inactive Task		Start-only	
	Split		Inactive Milestone		Finish-only	
	Milestone		Inactive Summary		Deadline	
	Summary		Manual Task		Critical	
	Project Summary		Duration-only		Critical Split	
	External Tasks		Manual Summary Rollup		Progress	
	External Milestone		Manual Summary			

Handwritten signature/initials

Handwritten signature/initials

Handwritten signature/initials in the top left corner.



Project: Разработване и внедряване Date: Mon 7.12.15	Task		Inactive Task		Start-only	
	Split		Inactive Milestone		Finish-only	
	Milestone		Inactive Summary		Deadline	
	Summary		Manual Task		Critical	
	Project Summary		Duration-only		Critical Split	
	External Tasks		Manual Summary Rollup		Progress	
	External Milestone		Manual Summary			

Handwritten signature/initials in the bottom left corner.

Handwritten signature/initials in the bottom right corner.

Предложение за интеграция на АСУД с единна среда за безхартиен обмен на дела на ВСС

за изпълнение на обществена поръчка по реда на Глава осма „а” от ЗОП с предмет „Оптимизация на бързодействието на Автоматизираната система за управление на делата АСУД и интегриране с единната среда за безхартиен обмен на дела във Върховния касационен съд”

Декларация по чл. 33 ал. 4 ЗОП

Декларация по ст. 33 ст. 4 ЗОК

Декларация по ст. 33 ал. 4 ЗОК

Декларация по ст. 33 ст. 4 ЗОК

Декларация по ст. 33 ст. 4 ЗОК

Предложение за оптимизация на модела на данни

за изпълнение на обществена поръчка по реда на Глава осма „а” от ЗОП с предмет „Оптимизация на бързодействието на Автоматизираната система за управление на делата АСУД и интегриране с единната среда за безхартиен обмен на дела във Върховния касационен съд”

Декларация по чл. 33 ал. 4 ЗОП

Декларация по ст. 33 ал. 4 ЗОК

Декларация по ст. 33 ст. 4 ЗОК




Приложение 5

Техническо описание на разработеното решение

за изпълнение на обществена поръчка по реда на Глава осма „а” от ЗОП с предмет „Оптимизация на бързодействието на Автоматизираната система за управление на делата АСУД и интегриране с единната среда за безхартиен обмен на дела във Върховния касационен съд”

Декларация по чл. 33 ал. 4 ЗОП

Декларация по ст. 33 ст. 4 ЗОК



Приложение 6

План за тестване

за изпълнение на обществена поръчка по реда на Глава осма „а” от ЗОП с предмет „Оптимизация на бързодействието на Автоматизираната система за управление на делата АСУД и интегриране с единната среда за безхартиен обмен на дела във Върховния касационен съд”

1. Въведение

Този документ описва подхода, средствата, видовете тестове и критериите, необходимите за провеждане на тестове, в рамките за изпълнение на обществена поръчка по реда на Глава осма „а” ОТ ЗОП с предмет „Оптимизация на бързодействието на Автоматизираната система за управление на делата АСУД и интегриране с единната среда за безхартиен обмен на дела във Върховния касационен съд”.

2. Подход за тестване

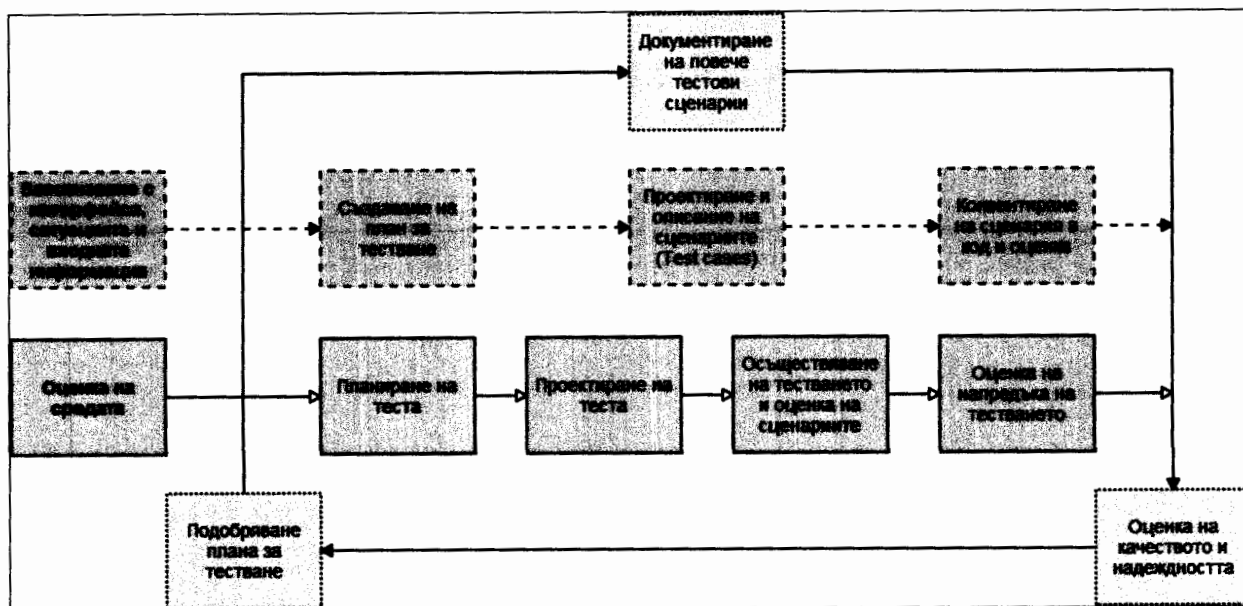
Подходът за тестване, който ще се използва включва елементи от въведените системи за управление на качеството по стандарт **ISO 9001:2008** в организацията на „Смарт системс 2010“ ЕООД и най-доброто от RUP подхода. На най-високо ниво целта на тестването е да потвърди, че разработеното решение отговаря 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 Изпълнителя ще следва:





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 Заключителни дейности.

2.1. 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, подмодули и компоненти на програмната система и идентифициране на критичните точки за бизнеса и отделните групи потребители на системата.

2.2. Анализ и проектиране

В етапа на Анализ и проектиране се определя последователността на тестовите и изискванията към тестовата среда, проектиране на тестовите 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 подчасти на програмната система с цел по-лесно проектиране на тестовите 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 подчасти 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.



2.3. Реализация и изпълнение

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.

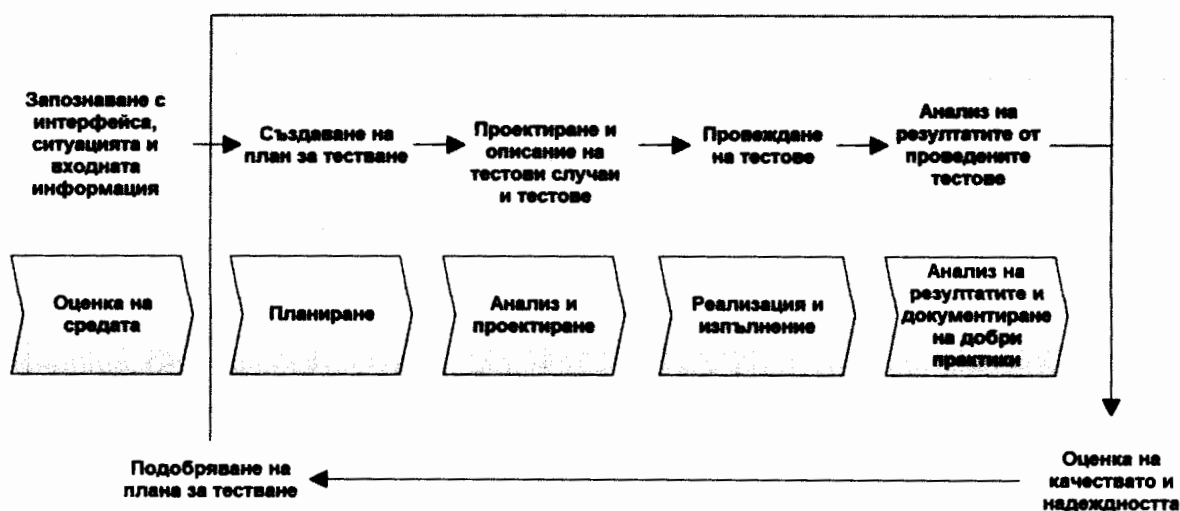
2.4. Анализ на резултатите

Етапът на Анализ на резултатите се състои от отчитане на получените резултати в избрания формат и проверка на условията за завършване на тестовете.

2.5. 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:



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.

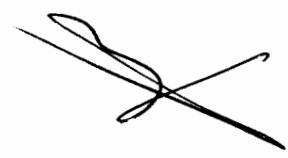
Тестова среда (Test harness)

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 подчасти на програмната система с цел по-лесно проектиране на тестовете чрез разделяне на множества, ориентирани към отделните модули;
- Идентифициране на основните групи тестове и тестови сценарии за интеракция 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;
 - Определя последователността на тестовете и изискванията към тестовата среда, проектиране на тестовете и подготовка на тестови данни
 - Определяне на критериите (изискванията към резултата от проведения тест) за преминаване/непреминаване на всеки тест.

Източници на информация:

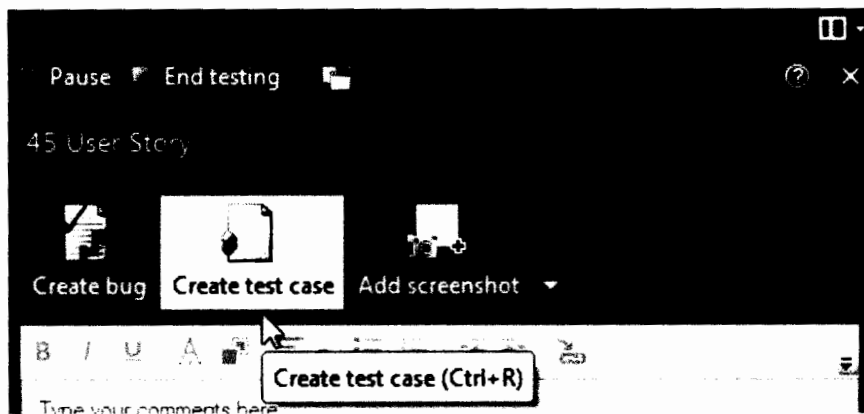
- **ISTQB** (International Software Testing Qualifications Board): <http://www.istqb.org/>;
- **iSQI** (The International Software Quality Institute): <https://www.isqi.org/>.

3. Средства за тестване

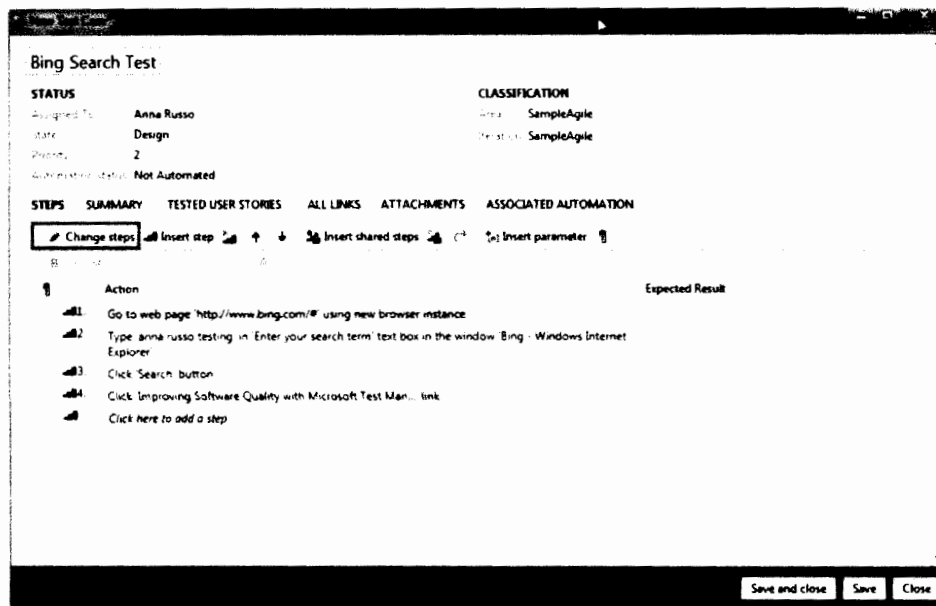
В процеса на тестване на всеки от разработените софтуерни компоненти Изпълнителят ще използва като основен инструмент **Visual Studio Team Foundation System (TFS)**. TFS е продукт на Microsoft, който предлага система за контрол над изходния код и неговите версии, събиране на информация, отчитане на извършената работа и следене на процеса на развитие на проект. TFS предоставя пълна функционалност за дефиниране на тестови сценарии, както и за разработване и изпълнение на всички видове тестове, които ще бъдат заложи в текущия проект.

VSTFS предоставя пълна функционалност за дефиниране на тестови сценарии, както и пълен набор от инструменти за проследяване и отстраняване на грешки (Bug Tracking System) при изпълнение на всички видове тестове, които ще бъдат заложи в текущия проект. Системата също така записва всички действия, извършени при проведените тестове, за да улесни идентифицирането на причините за възникналите грешки.





Създаване на тестови случай в Microsoft Visual Studio



Изглед на последователността на стъпки в тестови случай

TFS предлага няколко компоненти, свързани с груповата работа по разработката и тестване на проекта. В зависимост от предназначението те са разделени на следните видове:

- Контрол на версиите (Version control) – за управление на сорс кода и подобни работни продукти, които изискват поддръжка на история за различни версии;
- История на работни елементи (Work item tracking) – за пазене на история на елементи от работния процес, като дефекти, изисквания, задачи и сценарии за тестване;
- Функции за управление на проекти (Project management functions) – позволяват изготвянето на групов проект, базиран на дефинирано от потребителя софтуерно поведение. Също така те дават възможност да се планира и следи развитието чрез Microsoft Excel и Microsoft Project;
- Изграждане на работа в екип (Team build) – за създаване на общ екипен процес по изготвяне на работещ продукт;
- Събиране на информация и отчет (Data collection and reporting) – подпомага оценката на състоянието на груповия проект, базирана на информацията събрана от TFS инструментите за тестване;

- Портал на груповия проект (Team Project Portal) – предоставя централна точка за комуникация между екипа в рамките на Microsoft Windows SharePoint Services сайт;
- Team Foundation Shared Services – предоставят голям брой от елементи на вътрешната структура, които са невидими за крайните потребители, но важни за разработчиците на инструменти и разширения.

Visual Studio предлага използването на голям набор от инструменти като при комбинирането им с Visual Studio Team Foundation сървър могат да се въведат изпитани практики, чрез които да се ръководи цялостния процес на разработка и тестване на софтуера от конкретизация на нуждите на потребителя, през дизайн, архитектура, тестване и имплементиране на кода до крайно въвеждане в експлоатация.

Според това, каква цел екипът иска да постигне, инструментите на TFS могат да се групират според предназначението им, като инструментите, които имат отношение към провеждането на софтуерните тестове са:

Група	Предназначение
Подобряване на качеството с Visual Studio Diagnostic Tools (чрез използване на Version Control и Team Explorer Everywhere – Visual Studio 2010 и нагоре)	За разработка: от писане на код, unit тестове, тестване като цяло, дебъгване, анализиране до профилиране на приложението, използвайки приложения, които са интегрирани заедно с останалата част на средата и TFS.
Тестване на приложението	За изготвяне и извършване на „ръчни“ или автоматизирани тестове, включително тестове за производителност и натоварване.

Забележка: В зависимост от сложността на проекта и необходимостта от допълнително изпитване по време на етап Разработка Изпълнителят може да включи допълнителни инструменти за тестване на разработения софтуер с цел удовлетворяване на изискванията за работоспособност и сигурност на системата.

4. Видове тестове

С цел да се осигури качествен и отговарящ на нуждите от техническото задание продукт, софтуерът (портала и системата) ще бъде тестван обстойно. По-долу са описани видовете тестове, които екипът ни планира да изпълни:

4.1. Интеграционни тестове

За начало на интеграционното тестване се счита моментът, в който отделните компоненти/единици започнат да се тестват свързано. След като тестването на отделните компоненти на софтуера е завършило, отделните единици се интегрират във функционални групи и се тестват отново. Целта на този тип тестване е да се открият скрити дефекти и несъответствия в интерфейсите между отделните модули и обмена на информация с базите данни. Интеграционното тестване включва голям набор от дейности – като се започне от

тестването на няколко модула и се стигне до тестване на цялата система. Провеждането на интеграционни тестове е планирано за всяка итерация по време на етап Разработване на системата.

Цел:	Проверка на всички функционалности на системата, които имат взаимна обвързаност, за несъответствия в интерфейсите или скрити дефекти.
Техника:	За да се потвърди интегритета на функционалността на системата се изпълнява целия набор от написани тестови сценарии, като се използват валидни и невалидни данни за всеки един от параметрите.
Критерии за успешно завършване на тези тестове:	Няма намерени бъгове/дефекти в интеграцията на функционалните блокове
Очаквани резултати:	С провеждането на интеграционните тестове ще бъде предотвратено проявяването на скрити дефекти и несъответствия в интерфейсите между отделните модули и обмена на информацията с базите данни.

4.2. Тестове за функционалност (Functional Testing)

Тестовите за функционалността ще бъдат проведени с цел да удостовери пълнота по отношение на функционалността, която е заложена в разработката. Всички системни функционалности, изискващи въвеждане на данни ще бъдат тествани както с коректни, така и с некоректни данни. Добавянето на нова функционалност или поправянето на грешка в кода на разработката ще бъде предхождано от добавяне на нов тестов сценарий. Тестовите се изпълняват през графичния интерфейс на системата и на базата на подадените входни данни се прави анализ за изходните резултати. Вниманието на тестера е насочено към коректността на реализираните бизнес процеси и тяхното съответствие с функционалните изисквания към системата. Необходимо е да се използват валидни и невалидни входни данни.

Целта е да се установи дали:

- при валидни входни данни се генерират очакваните резултати;
- при невалидни входни данни системата генерира адекватни съобщения за грешка и не допуска потребителя до следващите нива на системата;
- навигацията между различните прозорци и полета е правилна;
- определени индустриални стандарти са спазени.

Цел:	Проверка на всички функционалности на портала на всеки един модул, включително навигация, входни данни, процеси и тяхното повторение.
Техника:	Изпълнение на целия набор от написани тестови сценарии, като се използват валидни и невалидни данни за всеки един от параметрите, за да е в състояние да се потвърди: <ul style="list-style-type: none">• очакваните резултати при правилно въведени данни според функционалната спецификация;

	<ul style="list-style-type: none">• визуализацията на информационни съобщения и състоянието на системата при неправилно/некоректно зададени входни данни;• всяка една потребителска роля за правилно ѝ използване в системата.
Критерии за успешно завършване на тези тестове:	<ul style="list-style-type: none">• всички предвидени и създадени тестови сценарии са изпълнени;• всички намери бъгове/дефекти са документирани.
Очаквани резултати:	С реализирането и провеждането на тестови сценарии за функционално тестване ще бъде анализирано и предотвратено проявяването на грешки свързани с въвеждането на входни данни, навигирането между различни прозорци, както и дали определени индустриални стандарти са спазени.

4.3. Тестове за натоварване (Stress Testing)

Тестовите за натоварване ще бъдат проведени през фазата на разработка на системата по време на системното тестване. При незадоволителен резултат на тестовите за натоварване източниците на проблема ще бъдат диагностирани и отстранени и ще бъдат осъществени подобрения за усъвършенстване на работата на системата.

Процесът на изпълнение включва въвеждането на големи стойности, които биха забавили достатъчно процеса по работа със системата, въвеждане на големи входящи/изходящи стойности гарантирайки зареждането на комплексни заявки към базата данни. По време на натоварването се наблюдавана коректната работа на системата, натоварването на процесор, памет и дали при тази натовареност системата поддържа наличност и коректност за други потребители (дали други потребители могат да работят със системата и се следи адекватността на отговорите).

Цел:	Проверка на системата според написаните тестови сценарии за функционални тестове. Тук ще се проверява системата в режим на натоварване, съгласно следните параметри: <ul style="list-style-type: none">• липса на достатъчно памет на сървърната машина (RAM);• максимален брой на свързаните клиенти (Connections);• извършване на едно и също действие, по едно и също време от две и/или повече регистрации.
Техника:	Използване на тестови сценарии, създадени за провеждането на Load тестове. Тестове трябва да бъдат извършени от една и съща машина, като паметта на сървъра трябва да бъде намалена (или лимитирана).

Критерии за успешно завършване на тези тестове:	Всички сценарии трябва да бъдат изпълнени, като се определят минимум ресурси за правилното функциониране на система без да се отчитат грешки в нея.
Очаквани резултати:	С изпълняването на тестови сценарии за натоварване ще бъде представена ясна и точна информация относно натоварването на системата при изпълнение на функционалности, като по този начин ще бъде предотвратено настъпването на системен срив или невъзможност за обработка на сложна структура от данни.

4.4. Тестове за производителност и работоспособност (Performance and Load Testing)

Тестове за производителност (Load testing)

Това е процес на създаване на заявки към системата или устройството и измерване на времето за отговор на съответната процедура. **Load testing** основно се отнася за очакваното използване на ресурсите от системата в процес на натоварване от множество потребители. Изследва се конкуренцията между отделните модули и процеси. Този тест се прилага при системи, подлежащи на много потребителско ползване, често при изграден клиент/сървър модел, също като уеб сървърите. Извежданите резултати от **Load testing** са свързват предимно с текущите и очакваните резултати, отколкото с теоретичните. Измерва се натовареността на системата за зареждане на отделните модули или компоненти в период на минимално или пиково натоварване.

Този тест се провежда успоредно с performance тестове.

Цел:	Проверка на състоянието на системата и функционалностите, които протичат при следните условия: <ul style="list-style-type: none">• нормална среда на функциониране на система;• натоварена среда на функциониране на система.
Техника:	Използване на тестови сценарии създадени за проверка на функционалностите на системата. Тук се увеличават до критичен максимум обема информация / елементи, които се подават към системата, посредством скрипт за провеждане на автоматизирани тестове. Провеждането на тези тестове трябва да се извърши първо от един регистриран потребител, след което да се повтори с множество симулирани заявки, посредством инструмент за автоматизирани тестове.
Критерии за успешно завършване на тези тестове:	Един потребител: успешно завършване на тестовия сценарий без отбелязването на грешки и идентично с очакваният резултат, според функционалната спецификация;

Очаквани резултати:	При изпълнение на тестови сценарии за проверка на производителност, ще бъдат анализирани използваните ресурси от системата в процес на натоварване от множество потребители, като по този начин ще се гарантира бързодействие при осъществяване на пиково натоварване на отделните модули или компоненти.
----------------------------	---

Тестове за работоспособност (Performance testing)

Задачата на тези тестове е да определяме дадени параметри от машината в процес на изпълнение на текуща процедура. **Performance Testing** е подкатегория на **Performance Engineering**, чрез която се стреми да се тества архитектурата и дизайна на системата, преди да се "пусне" официално, за да могат да се предвидят всички нейни действия в процес на натоварване от страна на крайният потребител.


Чрез провеждането на тези тестове се цели да се изследва състоянието на системата и да се изведат сравнителни данни между две такива, за да се избере най-подходящата конфигурация, да се изследва кои процеси натоварват най-много машината и кои биха предизвикали нейното "сриване". В процеса на диагностика, софтуерните инженери могат да използват *profilers* като мерни единици, за да разберат кои са най-критичните параметри от системата и да ги оптимизират.

При **Performance testing** се стреми да се симулират действия на потребители при които системата може да се натовари най-много. За провеждането на тези тестове е необходимо една или повече съвършни конфигурации, за да могат да се симулират определен брой потребители, изпълняващи определена последователност от действия. Всичко това е описано в предварително подготвени тестови скриптове, за да могат да се следи състоянието на зададените параметри от машината.

Първоначално се започва с малък брой потребители, като техният размер се увеличава пропорционално. Резултата от теста дава информация за представянето на системата в съотношение потребители - време за отговор. Могат да следят различни показатели измерващи състоянието на процесор, памет, *application servers*, *web services* и т.н.

Провеждането на **Performance Testing** обикновено се комбинира със **stress testing**, за да може да се разбере какво ще стане със система след като се превишат допустимите параметри. За да се получат пълни данни и да се изследва системата ефикасно, то трябва да се събере минимум информация:

- за обхвата на **Performance** теста – спецификацията на системата, интерфейса, компонентите и т.н. използвани за този тест;
- относно броят на потребителите, който да се очаква да извършват всяко едно действие - в минимум и в пиков момент;
- относно реакцията на системата, от гледна точка на хардуерна конфигурация и мрежова осигуреност
- за натовареността на работа на процесите при минимални и максимално натоварване, които ще се изследват - вход в системата, търсене, избор на опция, изход от системата;
- както и времевите изисквания за изпълнение на всяка операция при специфични условия и по норматив.



Цел:	Проверка на функционалностите на система според определено време за натоварване.
Техника:	Създаване на тестови сценарии, които да потвърдят правилното действие на проверяваната функционалност. Осъществяване на промяна за брой на симулираните заявки и времето за изпълнение на всяка една от тях. Тези тестове ще се извършват, чрез инструмент за провеждане на автоматизирани тестове
Критерии за успешно завършване на тези тестове:	Един потребител: успешно завършване на всички тестови сценарии без отбелязването на грешки и идентично с очакваният резултат, според функционалната спецификация; Множество симулирани заявки: успешно завършване на всички тестови сценарии без отбелязването на грешки и идентично с очакваният резултат, според функционалната спецификация. Документиране на всички открити грешки / дефекти.
Очаквани резултати:	С изпълнението на тестови сценарии за проверка на натоварване, и анализиранияте данни от тях, ще предоставят ясна представа за използваните ресурси от системата в процес на натоварване от множество потребители. По този начин ще се гарантира използването на максимален брой потребители в пиков момент на работа с отделните модули или компоненти.

4.5. *Тестове за удобство на работа и тестване на потребителския интерфейс (Usability Testing and User Interface Testing)*

Тестове за удобство на работа включват проверка дали изходът от програмата е смислен за потребителя, дали всички функционалности на системата са достъпни през потребителския интерфейс, дали системата дава незабавна обратна връзка при вход. Те се използват като оценка за разбираемостта на съобщенията за грешка, унифицираността на потребителския интерфейс от гледна точка на синтаксис, семантика, форматиране, стил и абревиатури и достатъчна прецизност на входа.

Тестове на потребителския интерфейс целят да проверят взаимодействието на потребителя със системата. Целта им е да осигурят, че интерфейсът предоставя на потребителя удобен достъп и навигация между функциите на системата. Също така тестването на потребителския интерфейс осигурява, че информацията, която се предоставя, е правилна и отговаря на изискванията.

С цел да се изследва удобството на работа със системата и степента на интуитивност на потребителския интерфейс ще бъде планирано допитване за мнението на потребители на системата чрез анкети.

Цел:	Проверка на следните параметри: <ul style="list-style-type: none"> • Навигацията в системата отговаря на бизнес логиката и
-------------	---




	логиката на процесите <ul style="list-style-type: none">Всички обекти отговарят на стандартите, включително менюта, размери, полета, позиции
Техника:	Провеждане на тестове за всеки един екран, за да се провери за размествания по екрана, размествания на обекти и полета, използвайки различни браузъри - Internet Explorer, Mozilla Firefox, Google Chrome и др.
Критерии за успешно завършване на тези тестове:	Всеки един екран отговаря на стандартите и изискванията. Липса на различия по дизайна, използвайки различни браузъри.
Очаквани резултати:	С провеждането на този вид тестове ще бъдат проверени взаимодействието на потребителя със системата, като по този начин ще бъде осигурен интерфейс, който предоставя на потребителя удобен достъп и навигация между функциите на системата.

4.6. Тестове за контрол на сигурността и достъпа (Security Testing)

Тестването на сигурността се стреми да изчерпа всички възможни негативни сценарии. Процесът на тестването включва проверка на правата за достъп, възможност за авторизация и идентификация, криптиране на по-важните данни в базата. Използват се техники за успешно получаване на неправомерен достъп, като например атаки за блокиране на достъпа до даден ресурс. Проблемите, свързани със сигурността не зависят изцяло от пропуск в програмния код и имплементацията на системата. Средата и технологията за разработка също съдържат в себе си потенциални пролуки. Поради тази причина се извършват проучвания за откритите вече проблеми в конкретната среда, инсталират се поправки от производителя (patch) или се разработват такива, и се поставят допълнителни бариери пред нарушителите.

Цел:	Проверка за защита на функционалността и поверената информация в системата.
Техника:	Провеждане на тестове за неправомерно достъпване на информация или намеса във функционалността на системата.
Критерии за успешно завършване на тези тестове:	Всеки един от използваните методи, не трябва да достъпва неправомерно информация или да променя ключова функционалност. Всяка външна намеса се води за провал в сигурността на системата.
Очаквани резултати:	Този вид тестови сценарии ще провери процеса на проверка на правата за достъп, възможност за авторизация и идентификация, криптиране на по-важните данни в базата, като с използваните техники ще се предотврати получаването на неправомерен достъп до чувствителна информация.



4.7. Тестове за възобновени дефекти (Regression Testing)

Регресионното тестване (Regression testing) представлява повторно тестване на програмата, след нейното модифициране, за да се провери, че не са настъпили грешки в модулите, които не са претърпявали промяна. Регресии могат да се наблюдават и при промяна на конфигурацията на системата. Преди предаване на нова/актуализирана версия на Възложителя или след отстраняване на регистрирани грешки, системата следва да бъде тествана отново, за да се провери дали не са възникнали проблеми при съществуващите функции.

Важна предпоставка за този тип тестове е анализът на влиянието на промените – определяне кои точно части на системата ще бъдат засегнати от промените. Тестовите трябва да проверят дали:

- непроменените части от системата имат непроменено поведение;
- модифицираните части работят както се очаква;
- системата като цяло изпълнява изискванията.

С цел улесняване на този вид тестване е прието изготвянето и автоматизацията на набор от основни тестове, обхващащи функционалността на системата заедно с тестове, описващи наблюдавани, но вече поправени програмни дефекти.

Цел:	Проверка на документираните бъгове / дефекти за тяхното отстраняване.
Техника:	Използване на тестовите сценарии, създадени за провеждането на функционалните тестове.
Критерии за успешно завършване на тези тестове:	Потвърждаване на следните параметри: <ul style="list-style-type: none">• Всички предвидени и създадени ТС са изпълнени• Всички документираните бъгове / дефекти са ретествани. При наличие на репродуциране на грешки, те отново се документират.
Очаквани резултати:	С провеждането на този вид тестове ще бъде гарантирано, че в процеса на имплементиране на нови функционалности няма да възникнат дефекти свързани с непроменени функционалности.

4.8. Тестове за приемане на системата (Acceptance Testing)

Този тип тестване има за цел да потвърди, че разработената система отговаря на нуждите на Възложителя. След изготвянето на критериите за приемане на програмната система (acceptance criteria) те се обсъждат с възложителя. На базата на договорените критерии за приемане на системата тест екипът подготвя тест сценарии за приемане на системата. Тези сценарии (acceptance scenarios) биват представени и одобрени от оторизиран представител на възложителя и след това стават база за проектиране на приемателните тестове (acceptance testing). Тестове за приемане на системата се провеждат едва след като софтуерът премине успешно всички системни тестове и бета тестване.



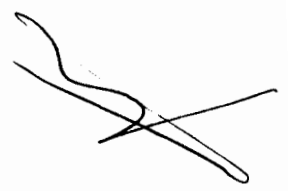
Цел:	Потвърждаване на следните параметри: <ul style="list-style-type: none">• резултатите от тестването са успешни;• системата отговаря на изискванията заложи в техническото задание.
Техника:	Провеждат се тестове разработени по изискване на Възложителя за оперативност на системата. Резултатите от тези тестове удостоверяват на Възложителя, че системата е в готовност за пускане в реална експлоатация.
Критерии за успешно завършване на тези тестове:	Всеки един екран и отговор на системата отговаря на стандартите и изискванията, както и липса на различия по дизайна, използвайки различни браузъри.
Очаквани резултати:	След окончателно разработване системата ще бъде демонстрирана на Възложителя и с помощта на предварително изготвените тестови сценарии ще бъде удостоверено коректното функциониране на системните функционалности спрямо одобрената техническа спецификация.

Забележка: Обхватът на тестовите ще бъде конкретизиран по време на етап Разработване, след което по преценка на Изпълнителя някои от гореизброените видове тестове може да отпаднат, а други да бъдат добавени.

5. Критерии за приемане на проектните резултати

Разработваните функционалности по настоящата обществена поръчка ще се приемат в експлоатация само ако:

- удовлетворяват писмено поставените изисквания на Възложителя;
- успешно са преминали всички дефинирани тестове;
- не произвежда неправилни резултати (и не изпада в състояние да не произведе резултат) при правилни входни данни;
- не нарушава целостта си и целостта на съхраняваната информация в следствие на некоректни входни данни, програмни или други грешки;
- не изпада в недетерминирани състояния ("блокира", "заспива") в следствие на некоректни входни данни, програмни и други грешки или продължителна работа;
- няма синтактични и логически грешки;
- няма грешки или съществен спад (по-голям от 30%) на производителността, проявяващи се по време на претоварване, увеличаване на капацитета на БД или автоматични действия по архивиране, индексване и т.н.;
- няма грешки, зависещи от настъпването и взаимодействието на асинхронно възникващи събития, както и от забавянето на отговора/реакцията на други приложения;
- няма грешки, проявяващи се след системен срив или системно аварийно възстановяване след изключителни събития (напр. отпадане на захранването или апаратна повреда);



- документацията му е без грешки и неточности;
- съществуват показания, че функционалността може да обработи значително по-големи обеми данни без значителен спад на производителността;
- функционалността произвежда резултат в очакваното време за отговор.

