

Приложение 2

към




ТЕХНИЧЕСКО ПРЕДЛОЖЕНИЕ

за участие в открита процедура за възлагане
на обществена поръчка с предмет:

**„Доставка на интегрирана система за корабна инспекция –
софтуерен пакет с електронна база данни и оборудване“**

Предложение за организация за
изпълнение и управление на поръчката

2018 г.



000119

Съдържание

1. Въведение..... 5

2. Методология за управление на проекта 5

2.1. Общ подход за управление на проекта 5

2.2. Процеси за управление на проекта съгласно PMBOK 6

2.3. Процесни области при управление на проекта 6

2.3.1. Управление на интеграцията 6

2.3.2. Управление на обхвата 7

2.3.3. Управление на времето 8

2.3.4. Управление на разходите 8

2.3.5. Управление на качеството 9

2.3.6. Управление на човешките ресурси 10

2.3.7. Управление на комуникацията 10

2.3.8. Управление на риска 10

2.3.9. Управление на заинтересованите страни (stakeholders) 11

2.3.10. Управление на доставките 11

2.4. Общи управленски дисциплини 12

2.5. Общи организационни принципи 12

2.6. Обосновка за избора на предложената методология 13

3. Предварителен план за управление на проекта 13

3.1. Организация на работа 13

3.1.1. Организация на взаимодействието между Изпълнителя и Възложителя 13

3.1.2. Органи за управление на проекта 14

3.1.3. Организационна диаграма 16

3.1.4. Роли и отговорности 16

3.1.5. Участие на експертите в дейностите по попоекта 21

3.1.6. Координация в екипа на Изпълнителя 22

3.2. Управление на комуникацията 22

3.2.1. Проектна комуникация 22

3.2.2. Правила и канали за комуникация 22

3.2.3. Управление на документацията и инструменти за комуникация 24

3.2.4. Провеждане на срещи 25

3.2.5. Канали за комуникация 26

3.2.6. Докладване 26

3.3. Управление на качеството 28

3.3.1.	Стратегия за качество в проекта	28
3.3.2.	Процедури за контрол на качеството.....	29
3.3.3.	Роли и отговорности в процесите по осигуряване и контрол на качеството	29
3.4.	Мерки за мониторинг и контрол	31
4.	Етапи и дейности в обхвата на поръчката	34
4.1.	Описание на дейностите	34
4.2.	Резултати от изпълнението по етапи и дейности	35
4.3.	Ресурсна обезпеченост.....	40
5.	Описание на подход и методи за ключови дейности от изпълнението на проекта.....	41
5.1.	Подход за бизнес анализ.....	41
5.2.	Подход за проектиране на софтуер.....	41
5.2.1.	Процес за проектиране	41
5.2.2.	Софтуерна архитектура.....	45
5.2.3.	Обектно-ориентиран дизайн	47
5.2.4.	Моделиране с UML	48
5.2.5.	Софтуер за моделиране на UML – Sparx Systems Enterprise Architect	50
5.2.6.	Адаптиране на подхода за проектиране на софтуер	51
5.3.	Подход за софтуерна разработка	51
5.3.1.	Процес за софтуерната разработка	51
5.3.2.	Характеристики на процеса за софтуерна разработка	56
5.3.3.	Вътрешно-екипна координация в рамките на итерациите	57
5.3.4.	Обектно-ориентирано програмиране	59
5.3.5.	Дейности за интегриране на функционалности/компоненти в системата.....	60
5.3.6.	Стратегия за интегриране на функционалности/компоненти в системата	63
5.3.7.	Дневни билдове и тестове	64
5.3.8.	Управление на версиите.....	64
5.3.9.	Описание на инструментите, които ще се ползват от изпълнителя за извършване на разработка, тестване и внедряване	68
5.3.10.	Адаптиране на подхода за софтуерна разработка.....	69
5.4.	Подход за тестване	69
5.4.1.	Стратегия за тестване.....	69
5.4.2.	Начин на провеждане на тестовете.....	71
5.4.3.	Видове тестове	71
5.4.4.	Входни и изходни критерии	73
5.4.5.	Докладване на дефекти и несъответствия	74
5.4.6.	Инструменти за провеждане на тестовете	75

Handwritten signature

5.4.7.	Адаптиране на подхода за тестване.....	75
5.5.	Подход за внедряване.....	76
5.5.1.	Общи положения.....	76
5.5.2.	Стъпки за внедряване.....	76
5.6.	Механизъм и процедури за включване на граждани в процесите по разработка, тестване и идентифициране на пропуски на софтуера.....	77
6.	Документация.....	78
6.1.	Изисквания към документацията.....	78
6.2.	Прозрачност и отчетност.....	78
6.3.	Спецификация на софтуерните изисквания.....	79
6.4.	Техническа документация.....	79
6.5.	Протоколи.....	79
6.6.	Комуникация и доклади.....	79
7.	Предпоставки за успешно изпълнение на поръчката.....	79

1. Въведение

Настоящото приложение към техническото предложение описва предложената организация за управление и изпълнение на поръчката, като включва:

- Методология за управление на проекта;
- Предварителен план за управление на проекта;
- Етапи, срок и график за изпълнение;
- Описание на подход и метод за ключови дейности от изпълнението на проекта;
- Обучение;
- Документация.

Представена е концепция, в която са описани всички етапи и действия по изпълнение на услугите и дейностите на поръчката, включително методите за тяхното осъществяване и резултатите, които ще бъдат постигнати от реализацията им.

Предложена е организация на работа, която показва разпределение на задачите и отговорностите между отделните експерти; показани са взаимовръзките между тях и способите за взаимодействие с възложителя ИАМА.

Предложената концепция съответства на предмета и обхвата на поръчката, разработена е съобразно структурната, смислова и логическа последователност на Техническата спецификация на възложителя и би осигурила необходимото качествено ниво на изпълнение.

Представен е план-график за управление на изпълнението, който е съобразен с времетраенето и последователността на всеки етап.

Предложени са механизми за проследимост и отчетност на изпълнението, които да осигурят ефективно управление на проекта, така че да се гарантира степента на постигане на целите при съпоставяне на действителните и очакваните резултати от изпълнението.

2. Методология за управление на проекта

2.1. Общ подход за управление на проекта

При управлението на проекта ще бъде използвана практикуваната при нас система от правила, принципи и стандарти на Института за Управление на Проекти (Project Management Institute, PMI), систематизирани в Ръководство за Система от Знания за Управление на Проекти (PMBOK Guide) – пето издание.

2.2. Процеси за управление на проекта съгласно PMBOK



Управлението на проекта представлява прилагане на знания, умения, средства и методи в дейностите, за да се изпълнят изискванията на проекта. Това изисква ефективно управление на съответните процеси. Процесите за управление на проекта са групирани в пет групи:

- Група процеси за инициране – тези процеси се провеждат с цел дефиниране на нов проект или фаза на съществуващ проект за получаване на разрешение за започване на проекта или фазата.
- Група процеси за планиране – тези процеси се изпълняват за установяване на обхвата на проекта, прецизиране на целите, дефиниране на курс на действие, необходим за постигане на тези цели.
- Група процеси за изпълнение – тези процеси се изпълняват за реализиране на определената работа в плана за управление на проекта, за да бъдат спазени спецификациите на проекта.
- Група процеси за наблюдение и контрол – тези процеси са необходими за проследяване, контрол и регулиране на напредъка и изпълнението на проекта, идентифициране на области, където са необходими промени в плана и инициране на съответните промени.
- Група процеси за приключване – процеси, които се изпълняват с цел финализиране на всички дейности от всички групи процеси за формалното приключване на проекта или дадена фаза от него.

2.3. Процесни области при управление на проекта


2.3.1. Управление на интеграцията

Управлението на интеграцията в проекта включва процеси и дейности, нужни за идентифициране, дефиниране, комбиниране, унифициране и координиране на разнообразни процеси и дейности за управление на проекта в рамките на групите процеси за управление на проекта. В рамките на контекста за управление на проекти, интеграцията включва характеристики като унификация, консолидация и интеграционни дейности, които са решаващи за завършването на проекта и успешното управление на очакванията на заинтересованите страни и покриване на изискванията. Управлението на интеграцията в проекта изисква вземането на решения относно разпределянето на ресурсите, балансиране между конкуриращи се цели и алтернативи и управлението на зависимости във всички области на управление на проекта.

Процеси при управление на интеграцията:

- Разработване на Проектна дефиниция (Харта на проекта) – процес на разработване на документ, който формално оторизира проекта или фаза от него и документира началните изисквания, които ще удовлетворят нуждите и очакванията на заинтересованите страни.
- Разработване на План за управление на проекта – процесът на документиране на действията, необходими за дефиниране, подготовка, интегриране и координиране на всички допълнителни планове.
- Направляване и управление изпълнението на проекта – процесът на изпълнение на работата, дефинирана в плана за управление на проекта за постигане на целите на проекта.
- Наблюдение и контрол на работата по проекта – процесът на проследяване, преглед и регулиране на напредъка с цел покриването на целите за производителност, дефинирани в плана за управление на проекта.
- Интегриран контрол на промените – процесът на преглед за всички искания за промяна, одобряване на промени и управление на промените към отчетните продукти, проектните документи, проектните активи на организацията и плана за управление на проекта.
- Затваряне на проекта или фаза от проекта – процесът на финализиране на дейностите в група процеси за управление на проекта с цел формалното приключване на проект или фаза на проект.


Очаквани резултати:

- Проектна дефиниция (Харта на проекта);
 - План за управление на проекта;
 - Изпълнени работни пакети (задачи) по проекта;
 - Данни за напредъка и производителността;
 - Заявки за промени и решения по тях;
 - Научени уроци;
 - Формално приключени фази и проект.
- 

2.3.2. Управление на обхвата

Управлението на обхвата в проекта включва процесите, необходими за осигуряване на това, че проектът включва цялата изисквана работа и само работата, която се изисква за успешното му приключване. Управлението на обхвата най-вече се отнася до определяне и контролиране на това какво се включва и какво не в проекта.

Процеси при управление на обхвата:

- Събиране на изисквания – процесът на дефиниране и документиране на нуждите на заинтересованите страни за постигане на проектните цели.
 - Дефиниране на обхвата – процесът на разработване на детайлно описание на проекта и продукта.
 - Създаване на структура на пакетите от работа (WBS) – процесът на разделяне на проектните отчетни продукти и работа на по-малки и управляеми части.
 - Проверка на обхвата – процесът на формалното приемане за завършените отчетни продукти в проекта.
 - Контрол на обхвата – процесът на мониторинг на статуса на обхвата на проекта и продукта, и управление на промените.
- 

Тези процеси си взаимодействат с всички останали процеси в другите области при управлението на проекта. Всеки от процесите може да изисква усилия от един или повече хора в зависимост от конкретните нужди в проекта. Всеки процес се появява поне веднъж в проекта и веднъж или повече в отделните фази на проекта, в случай че проектът е разделен на фази.

Очаквани резултати:

- Специфицирани изисквания и нужди на заинтересованите страни;
- Дефиниция на обхвата;
- Структура на пакетите от работа (WBS).

2.3.3. Управление на времето

Управлението на времето в проекта включва изпълнението на задачи, необходими за навременното приключване на проекта.

Процеси при управление на времето в проекта:

- Дефиниране на задачите – процесът по идентифициране на специфични задачи, които следва да бъдат изпълнение за създаване на отчетните продукти по проекта.
- Задаване на последователност на задачите – процесът на идентифициране и документиране на зависимостите между задачите в проекта.
- Оценка на ресурсите по задачи – процесът на оценка на вида и количеството на материали, хора, оборудване или доставки, необходими за изпълнението на всяка задача.
- Оценка на продължителността на задачите – процесът на определяне на ориентировъчна продължителност на работното време, необходимо за изпълнение на всяка от задачите с определените ресурси.
- Разработване на график – процесът по анализиране на последователността на задачите, тяхната продължителност, изисквания за ресурси и ограничения с цел създаване на график за изпълнение на проекта.
- Контрол на графика – процесът на мониторинг на статуса на проекта за актуализиране на напредъка и управление на промените спрямо приетия график.

Очаквани резултати:


- Дефинирани на задачите;
- Последователност и зависимост на задачите;
- Оценка на ресурсите по задачи;
- Продължителност на задачите;
- График.

2.3.4. Управление на разходите


Управлението на разходите в проекта включва оценка, бюджетиране и контролиране на разходите така, че проектът да бъде завършен в рамките на одобрения бюджет.

Процеси:

- Оценка на разходите – процесът на разработване на ориентировъчна оценка на паричните средства за изпълнение на проектните задачи.
- Определяне на бюджет – процесът на обобщаване на оценените разходи за отделните задачи или пакети от работа за определяне и оторизиране на одобрен план за разходите (включващ разпределението им във времето).

- 
- Контролиране на разходите – процесът на мониторинг на статуса на проекта за актуализиране на бюджета на проекта и управление на промените в приетия план за разходите.


Очаквани резултати:

- Оценка на разходите;
 - Бюджет.
- 

2.3.5. Управление на качеството

Управлението на качеството в проекта включва процесите и дейностите на изпълнителя, които определят политиките за качество, целите и отговорностите така, че проектът да удовлетвори нуждите, поради които е предприет. Това включва системи за управление на качеството, политики за качество и процедури за постоянно подобряване на процесите.

Процеси при управление на качеството:

- Планиране на качеството – процесът по идентифициране на изискванията за качество и/или стандартите, на които трябва да отговаря проекта и продукта, и документиране на това как тези стандарти и изисквания ще бъдат прилагани в проекта.
 - Осигуряване на качеството – процесът на одититане на изискванията за качество и резултатите от измерванията за контрол на качеството, осигуряващи приетите стандарти. Осигуряването на качеството се прилага спрямо процесите в проекта.
 - Контрол на качеството – процесът на мониторинг и регистриране на резултатите от изпълнение на дейностите по контрол на качеството за оценка на производителността и препоръчване на промени. Контролът на качеството се прилага спрямо продуктите.
- 

Управлението на качеството се отнася към проекта и към продуктите от проекта. То се прилага във всички проекти, независимо от естеството на техния продукт.


Очаквани резултати:


- План за управление на качеството;
- Осигурено ниво на качеството на продукта или услугата, разработвани в проекта;
- Осигурено спазване на процесите в проекта.

2.3.6. Управление на човешките ресурси


Управлението на човешките ресурси включва процесите на организиране, управление и лидерство на проектния екип. Проектният екип включва хора с определени роли и отговорности за изпълнение на проекта. Видът и броя на членовете на екипа може да се променя често в хода на изпълнение на проекта. Въпреки, че членовете на екипа имат възложени конкретни роли и отговорности, въвличането на всички членове при планирането и вземането на решения може да бъде полезно. Ранното включване и участие на членовете на екипа добавя техния опит в процеса на планиране и подсилва тяхната ангажираност към проекта.

Процеси при управление на човешките ресурси:

- Разработване на План за управление на човешките ресурси – процесът на идентифициране и документиране на ролите и отговорностите в проекта, необходимите умения и познания на членовете на екипа, йерархията и организацията в екипа и създаването на план за управление на персонала.
 - Набиране на проектен екип – процесът на потвърждаване на наличността на човешките ресурси и присъединяването им към екипа на проекта. Формално назначаване в проектния екип.
- 

- 
- Развиване на проектния екип – процесът по усъвършенстване на компетентностите, взаимодействието в екипа и цялостната среда в екипа с цел подобряване на производителността в проекта.
 - Управление на проектния екип – процесът по проследяване на производителността на членовете на екипа, даване на обратна информация, разрешаване на казуси и управление на промени за оптимизиране на производителността в проекта.


Очаквани резултати:

- План за управление на човешките ресурси;
 - Формиран екип за изпълнение на проекта.
- 

2.3.7. Управление на комуникацията

Управлението на комуникацията в проекта включва процесите, необходими за осигуряване на навременно и точно генериране, събиране, разпространение, съхранение и структуриране на проектната информация. Проектните ръководители прекарват значителна част от времето си в общуване с членовете на екипа и заинтересованите лица – външни или вътрешни за организацията.

Процеси:

- 
- Планиране на комуникацията – процесът на определяне на нуждите от информация на заинтересованите лица и определяне на подхода за комуникация.
 - Разпространяване на информация – процесът по предоставяне на информацията на заинтересованите лица по проекта така, както е планирано.
 - Управление на очакванията на заинтересованите лица – процесът на комуникация и работа със заинтересованите лица за посрещане на техните нужди и обсъждане на казуси при тяхната поява.
 - Докладване на производителността – процесът на събиране и разпространяване на информация за производителността, включително доклади за състоянието и измерване на напредъка и прогнози.


Очаквани резултати:

- План за комуникация;
- Артефакти (доклади, протоколи от работни срещи и др.) от комуникацията.

2.3.8. Управление на риска

Управлението на риска в проекта включва процесите по планиране управлението на риска, идентификация и анализ, планиране на ответни действия, мониторинг и контрол на рисковете в проекта. Целта на управлението на риска е повишаване на вероятността и влиянието от позитивните събития и понижаване на вероятността и влиянието на негативните събития върху проекта.

Процеси при управление на риска в проекта:

- Планиране на управлението на риска – процесът на дефиниране как ще бъдат управлявани дейностите по управление на риска в проекта.
 - Идентифициране на рисковете – процесът по определяне на рискове, които могат да въздействат на проекта и документиране на техните характеристики.
 - Качествен анализ на рисковете – процесът по приоритизиране и категоризиране на рисковете, както и оценяване на тяхната вероятност и влияние.
 - Количествен анализ на рисковете – процесът по количествено анализиране на ефекта от идентифицираните рискове върху целите на проекта.
- 

- Планиране на ответни действия – процесът на разработване на опции и действия за подсилване на положителните възможности и намаляване на заплахите спрямо целите на проекта.
- Наблюдение и контрол на рисковете – процесът по изпълнение на планираните ответни действия, регистриране и проследяване на идентифицираните рискове, мониторинг на остатъчните рискове, идентифициране на нови рискове и оценка на ефективността на процесите за управление на риска.

Очаквани резултати:

- План за управление на риска;
- Регистър на рисковете;
- Качествен и количествен анализ на рисковете;
- Планове за ответни действия при възникване на рискове.

2.3.9. Управление на заинтересованите страни (stakeholders)

Процеса по управление на заинтересованите страни започва с идентифицирането на тези хора и групи от организацията, които имат отношение към или в следствие на проектната реализация.

Идентифициране на заинтересованите лица – процесът на идентифициране на всички хора и организации, повлияни от проекта и документирането на информацията за техните интереси, отношение и влияние върху успеха на проекта.

Основната полза от управлението на този процес, е че позволява на Ръководителя на проекта да идентифицира най-подходящия фокус към всяка една от заинтересованите страни или група от хора имащи отношение към проекта.

Очаквани резултати:

- Регистър на заинтересованите лица.

2.3.10. Управление на доставките

Управлението на доставките в проекта включва процесите, необходими за доставка или придобиване на продукти и услуги извън проектния екип.

Процеси при управление на доставките:

- Планиране на доставките – процесът на документиране на решенията за доставки в проекта, специфициране на подхода и идентифициране на потенциалните доставчици (в конкретния случай доставчиците са определени).
- Провеждане на доставките – процесът по набиране на оферти от доставчици, избор на доставчик и сключване на договор (в конкретния случай набиране на оферти няма да е необходимо).
- Администриране на доставките – процесът по управление на взаимоотношенията при доставка, мониторинг на договора и правене на промени и корекции при необходимост.
- Приключване на доставките – процесът на завършване на всяка доставка в проекта. Включва и формалното приключване на договора.

Очаквани резултати:

- План за доставки;
- Документация за доставки;
- Извършени доставки.



2.4. Общи управленски дисциплини

Управление на конфигурацията – за продуктите, необходими за управление на конфигурацията, за които Възложителя не е посочил продукти или посочените от него продукти не са достатъчни, ще се използват продукти, интегрирани в единна система, посочени от Изпълнителя. Тези системи ще предоставят като минимум възможността за достъп до всички обекти в конфигурацията, маркиране на версиите и автоматизирано генериране на актуалната конфигурация и нейното съдържание. Достъп ще бъде осигурен на упълномощени от Възложителя лица в рамките на изпълнението на целия проект.


Управление на промените - спомага за оптимизиране на излагането на риск, неблагоприятното въздействие или прекъсването на нормалната експлоатация. Процесът включва подаване и регистриране на искания за промяна, оценка на въздействието, цената, ползите и рисковете, свързани с промяната, подготовка на бизнес обосновката и получаване на одобрение, управление и координация на изпълнението на промените, наблюдение и отчитане на дейностите по изпълнение на промените, затваряне и преглед на исканията за промяна. Исканията за промяна се подават само в писмен вид и могат да се инициират и от двете страни от упълномощени представители.


Изпълнителя ще използва специализирана система за целите на управление на промените. Ще бъде предоставено на Възложителя Детайлно описание на процедурата за управление на промените и достъп до системата. Записите за промените ще се преглеждат от ръководителите на Изпълнителя и Възложителя.

Управление на проблемите - включва грешки свързани с хардуер, софтуер и приложения. Идентифицирането на проблемите ще става с набор от автоматизирани и неавтоматизирани методи. Включва: контрол на проблема (идентифициране, регистриране, класифициране; проучване на причините; диагностика и идентифициране на мерки за решаването му; проследяване и наблюдение на проблемите), контрол на грешките (оценка и изпълнение на мерки за решаване на проблема; затваряне на проблема; проследяване и наблюдение на диагностицирани проблеми), предотвратяване на проблеми (анализ на тенденциите и извършване на насочени превантивни действия). Изпълнителят ще използва специализирани системи за целите на докладването на проблеми и управление на тяхното коригиране. Ще предостави детайлно описание на процедурата, по която ще управлява този процес, както и достъп до съответната система.

2.5. Общи организационни принципи


Задължително изискване е да се спазят утвърдените хоризонтални и вертикални принципи на организация на изпълнението на предмета на обществената поръчка за гарантирано постигане на желаните резултати от проекта, така че да се покрие пълният набор от компетенции и ноу-хау, необходими за изпълнение на предмета на поръчката, а също така да се гарантира и достатъчно ниво на ангажираност с изпълнението и проблемите на проекта:



- 
- Хоризонталният принцип предполага ангажиране на специалисти от различни звена, така че да се покрие пълният набор от компетенции и ноу-хау по предмета на проекта и същевременно екипът да усвои новите разработки на достатъчно ранен етап, така че да е в състояние пълноценно да ги използва и развива и след приключване на проекта;
 - Вертикалният принцип включва участие на експерти и представители на различните управленски нива, така че управленският екип да покрива както експертните области, необходими за правилното и качествено изпълнение на проекта, така и управленски и организационни умения и възможности за осъществяване на политиката във връзка с изпълнението на проекта. Чрез участие на ръководители на звената – ползватели на резултата от проекта, ще се гарантира достатъчно ниво на ангажираност на институцията с проблемите на проекта.

2.6. Обосновка за избора на предложената методология

Значителният обхват, кратките срокове на проекта и взаимодействието с множество заинтересовани страни изискват той да се управлява по стандартен унифициран начин. Затова сме избрали да приложим популярните добри практики и методология на световно утвърдена организация като Project Management Institute (PMI). Утвърдили сме добрите практики на PMI като основа за прилаганата от нас методология за управление на проекти. Екипът ни има необходимите познания и опит за прилагане на методологията, а проектният ръководител е професионалист с богат опит в областта с прилагането на този подход.



3. Предварителен план за управление на проекта

В настоящия предварителен план за управление на проекта са описани конкретни аспекти от проектното управление в съответствие с изискваната информация съгласно техническото задание. Представена е и информация, необходима за обективно оценяване на предложението. В случай, че бъдем избрани за изпълнител ще разработим подробен проектен план, обхващащ всички необходими области на проектното управление.

3.1. Организация на работа

3.1.1. Организация на взаимодействието между Изпълнителя и Възложителя

Опитът показва, че успешното изпълнение на ИТ проекти зависи от ефективната и мотивирана съвместна работа на множество хора, с различни роли, знания и отговорности. Всички те трябва да работят в синхрон за постигане на резултатите от проекта в предвидените срокове и с необходимото качество. За тази цел е необходимо проектирането на ефикасна организация със съответните правила, роли и отговорности. В следващите параграфи ще опишем нашето предложение за съставянето на една такава организация. Методологията, в началната фаза на проекта в тази организация следва да бъде преразгледана, детайлизирана и съгласувана съвместно с Възложителя.

Проектната организация описва всички основни роли и функции, заедно с техните права и задължения, които най-общо могат да бъдат разделени на няколко нива. Дефинирането и ефективното създаване на проектната организация, както и нейното последващо функциониране и контролиране се извършва незабавно след стартиране на проекта и е един от най-важните аспекти на първата фаза на всеки един проект. Препоръчителните роли и процеси в проектната организация, ще бъдат съгласувани между Възложителя и Изпълнителя след подписване на договор.

Управлението на проекта, съгласно най-добрите практики, се осъществява от организация, която е разпределена в три основни нива:



Strategicheskoto nivo: Това е най-високото ниво, което обикновено функционира под формата на т.нар. *Комитет за управление на проекта* и в него страните обикновено се представят от своите Управители, Изпълнителни директори или други оправомощени представители, които имат правото и отговорността да дефинират, както и да променят, целите и обхвата на проекта.

Управленско ниво: Това е оперативното управляващо ниво, в което страните са представени от своите *Ръководители на проекта*, които имат правото и отговорността да взимат всички управленски решения, касаещи постигането на дефинираните от Стратегическото ниво цели и резултати, при управление на предварително договорените обхват, срокове и ресурси.

Техническо ниво: Това е оперативното експертно ниво, в което влизат отговорните технически експерти, в лицето на *Ръководителите на екипите по проектиране и разработка* (обикновено софтуерни и/или бизнес архитекти), които носят правото и отговорността за проектирането на предложеното решение, което следва да бъде реализирано.

В духа на добрата практика при изпълнение на проекти с висока сложност или важност, структурата на проектната организация трябва да е напълно симетрична, като за всяко от основните нива на управление има по един отговорен ръководител, съответно от страна на Възложителя и на Изпълнителя.

3.1.2. Органи за управление на проекта

При съставянето на проектната организация следва да бъдат спазени както хоризонталният, така и вертикалният принцип на управление. Накратко, тези два принципа предполагат, че от една страна ще бъдат дефинирани вертикални нива на отговорност, и от друга – че за всяко ниво ще бъде номинирано лице, носещо подходящите права и отговорности за вземане на решения.

Право и отговорност на Възложителя е да номинира и включи в проектната организация лица с необходимите компетенции, управленски и организационни умения и възможности, и най-вече – със съответния авторитет и пълномощия, така че да се гарантират адекватен процес на взимане на решения, устойчивост на резултатите, дългосрочно развитие и непрекъснато подобрене.

Хоризонталният и вертикалният принцип на управление трябва да бъдат съблюдавани от Собствениците на проекта и от двете страни, както при формиране на оперативните екипи, така и при изграждане на вертикалната организация за ескалация, контрол и отчетност.

Ние предлагаме управлението на проекта да се осъществява на трите основни нива, посочени в предходната точка.

Комитетът за управление на проекта, като най-високо ниво в проектната организация, отговаря за определянето, контрола и спазването на стратегическите цели на проекта. Този орган съблюдава спазването на тези цели, както и взема стратегически решения и разрешава несъответствията възникнали при промяна на общите цели на проекта, включително има решаваща роля при управлението на рискове и промени с голямо влияние и/или изискващи допълнителни ресурси. Комитетът разрешава спорове и разногласия, които не са могли да бъдат решени на управленското организационно ниво. Решенията на комитета са окончателни.

Срещите на Комитета за управление на проекта ще се провеждат по график, съгласуван при стартиране на проекта и според сроковете зададени в Техническото задание и Проектният график. Решенията на Комитета ще бъдат задължителни за двете страни по проекта. При необходимост, среща на Комитета за управление на проекта може да бъде свикана извънредно от Собственика на проекта от страна на Възложителя или на Изпълнителя. По своя преценка членовете на Комитета за управление могат да поканят участници от другите нива, например Ръководителите на проекта и Техническите

ръководители, както и други заинтересовани лица. С оглед на постигане на синхрон в изпълнението на дейностите в различните администрации и общите цели на проекта, участието на представители на тези администрации в структурата на проектното управление е от ключово значение и е препоръчително за безпроблемно изпълнение на проекта.

Ръководителите на проекта, като управленско ниво в проектната организация, ще отговарят за вземане на всички текущи решения, касаещи реализацията на проекта, както разрешаването на спорове и разногласия, които не са могли да бъдат решени от по-ниското, експертно, организационно ниво. В случай на невъзможност даден проблем да бъде разрешен на това ниво, както и в случаите на въпроси и рискове, водещи до съществени промени в обхвата, графика или ресурсите на проекта, проблемът се ескалира към най-високото, стратегическо ниво.

Срещите на Ръководителите на проекта следва да се провеждат по график, съгласуван при стартиране на проекта, но не по-рядко от веднъж месечно. Решенията от срещите на Ръководителите на проекта ще бъдат задължителни за двете страни по проекта. При необходимост, среща на Ръководителите на проекта може да бъде свикана извънредно от Техническия ръководител от страна на Възложителя или на Изпълнителя. По своя преценка Ръководителите на проекта могат да поканят участници от другите нива, например Техническите ръководители и ръководителите на екипи, както и други заинтересовани лица, например Собствениците на проекта.

Техническите ръководители ще отговарят за вземане на всички оперативни решения, касаещи техническата реализация на внедряваните решения, изпълнението на задачите си по проекта, както и за разрешаване на спорове и разногласия, които не могат да бъдат решени вътрешно в дадения екип или между съответните два екипа, заети с конкретната технологична или проектантска задача. В случай на невъзможност даден проблем да бъде разрешен на това експертно ниво проблемът се ескалира към по-горното, управленско ниво.

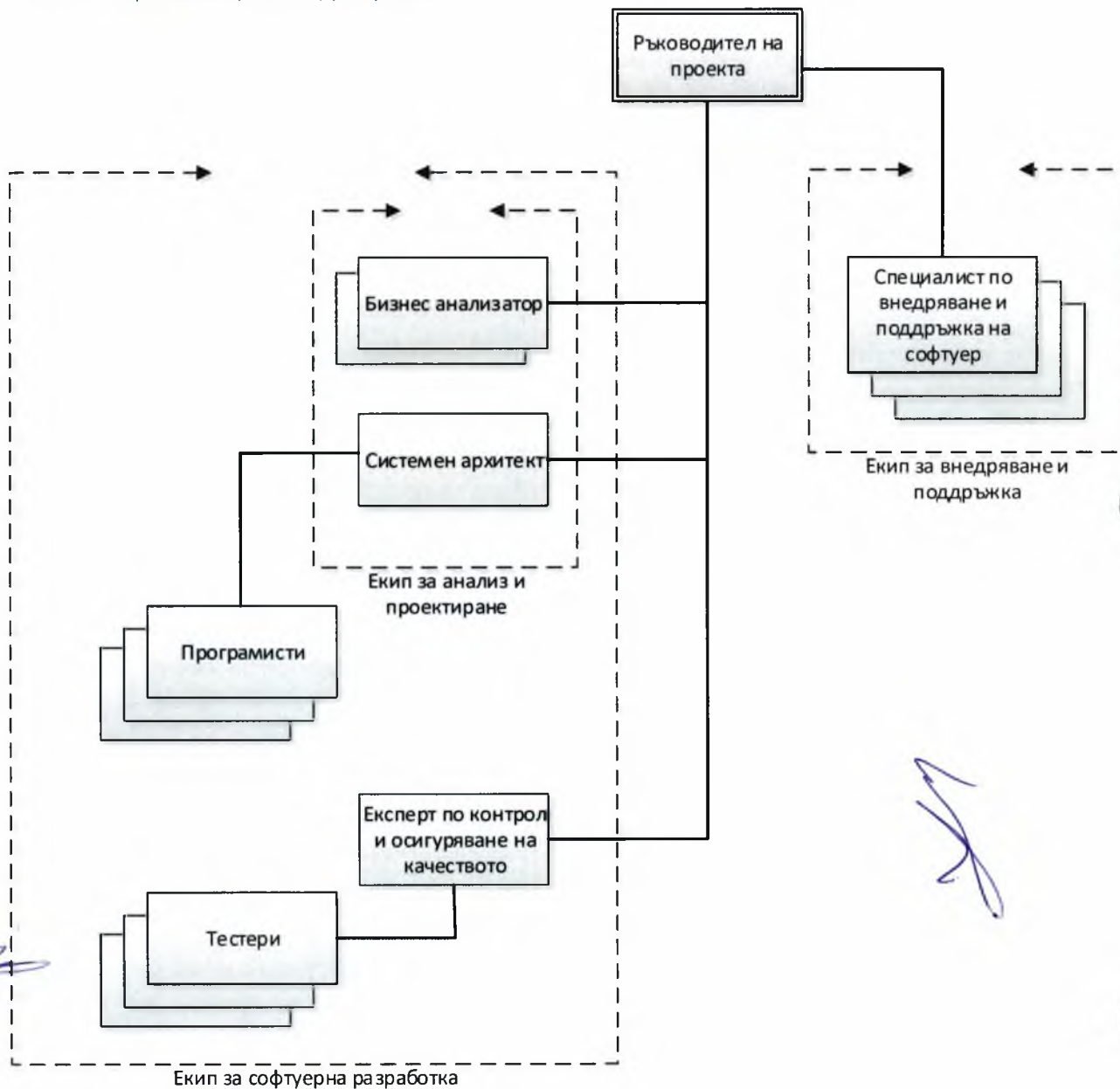
Препоръчително е Техническите ръководители да имат пълната свобода и отговорност да свикват технически срещи с честота и програма по своя преценка, като особено във фазата на анализ и проектиране това следва да е не по-рядко от веднъж седмично.

Когато трябва да бъде взето решение от по-високо ниво, т.е. наложително е даден проблем да бъде ескалиран, той се описва по възможно най-добрия начин и се изпраща според правилата за проектна комуникация до съответното по-високо ниво в организацията, както е описано по-горе.

Контролните органи по проекта ще могат да участват с в комитета за управление. Докладите за изпълнението на проекта ще бъдат разпространявани и до контролните органи по проекта. Проектат ще бъде управляван прозрачно и всяка необходима информация ще бъде предоставяна на контролните органи.

File

3.1.3. Организационна диаграма



3.1.4. Роли и отговорности

Роля	Отговорности
Ръководител на проекта (Ключов експерт)	Отговаря за успешното изпълнение на проектите, които управлява; Отговаря за подготовката на изпълнението на проекта, включително: първоначална идентификация на рисковете; изготвяне на план за комуникация; идентификация и план за управление на заинтересовани лица; Отговаря за планиране и ре-планиране на дейностите по проекта (ресурси, план-график); Мониторинг и оценка на изпълнението на проекта (по отношение на срокове, бюджет и други дефинирани показатели);

File

Роля	Отговорности
	<p>Контрол по изпълнението на проекта (иницииране или изпълнение на корективни действия);</p> <p>Отговорен е за управлението на рисковете свързани с работата по проекта (включително идентифициране, оценка, регистриране, предотвратяване или смекчаване на последствията при настъпване);</p> <p>Отговорен е за управлението на проблемите свързани с работата по проекта (включително идентифициране, оценка, регистриране и действия по разрешаване);</p> <p>Отговорен е за управление на промени и обхвата на проекта;</p> <p>Отговорен е за управление взаимоотношенията и комуникацията с всички заинтересовани лица (stakeholders), свързани с реализацията на проекта (напр. представители на клиента, партньори, под-изпълнители, ръководството на фирмата, екипа по изпълнение и т.н.);</p> <p>Отчитане на статуса (напредъка) по проекта - вътрешно (във фирмата) и външно (пред Възложителя), включително участие в съвет за управление на проекта;</p> <p>Участва в дейностите по одобрение на отчетената работа по проекта и месечното приключване;</p> <p>Отговаря по навременното изпълнение на всички административни дейности по обслужване на договора;</p> <p>Приключването на проекта – изготвянето на оценка от изпълнението на проекта, извличане и споделяне на научените уроци;</p> <p>Отговаря за дефинирането и изпълнението на индикатори за проектите, които управлява.</p>
<p>Бизнес анализатор (Ключов експерт)</p>	<p>Ролята на Бизнес системния анализатор се изразява най-общо в „Определяне, проучване, анализиране, систематизиране и документиране на дейностите, процесите и нуждите на клиента“. Той е посредникът между бизнес потребителите и техническия екип при разработката на софтуер.</p> <ul style="list-style-type: none"> • Работи съвместно с потребителите и екипа по разработка за определяне на оптималния за двете страни подход по отношение на изискванията, които са в рамките на проекта. Подпомага активно управлението на обхвата на проекта. • Документира резултатите от направените проучвания и анализи посредством изготвяне на функционални и други спецификации, както и изготвянето на прототипи на потребителския интерфейс. Актуализира изготвените спецификации през целия жизнен цикъл на проекта. • Представя резултатите от анализа пред потребителите и екипа по разработка, като подпомага двете страни в постигането на общо и пълно разбиране за бизнес изискванията. • Участва в процеса по промяна на изискванията, като обновява съответната документация и осигурява достигане на информацията за промяната до заинтересованите членове на екипа. • Подпомага екипа по поддръжка в изготвяне на потребителска документация и участва в провеждането на обучения за потребители. • Работи съвместно с екипа по осигуряване на качеството за осигуряване на съответствие на продукта с поставените изисквания, вкл. подпомага изготвянето на план за тестване и участва в тестове по приемане. • Контактна с екипа по осигуряване на качеството и екипа по разработка, като следи в хода на проекта за точното изпълнение на изискванията и проактивно взема мерки при установяване на несъответствия между поставените изисквания и реализираните резултати.

Handwritten signature

Роля	Отговорности
	<ul style="list-style-type: none">Извършва оценка на резултатите от тестовете на базата на основните качествени показатели, нормативни изисквания, както и изискванията и очакванията на клиента.
Системен архитект (Ключов експерт)	<p><u>Проектни дейности (в ролята на ръководител на екип по проект):</u></p> <ul style="list-style-type: none">Отговаря за навременното и коректно отчитане и одобрение на работата и дейностите, извършвани от хората в неговия екип;Отговаря за вътрешната организация и координация (работни процеси) на работата в екипа при изпълнението на проекти, както и за координирането на работата и резултатите на екипа с Ръководителя на проекта;Отговорен е за идентифицирането и предотвратяването на рисковете, свързани с работата на екипа. Ескалира рисковете към Ръководителя на проекта;Отговаря за дефинирането и изпълнението на цели за качество за екипа, който ръководи. <p><u>Технически дейности:</u></p> <ul style="list-style-type: none">Верифициране и приемане на изискванията, предоставени от Бизнес анализатора;Проучва технологии и варианти за решения;Технически анализ на изискванията. Измисляне на технически решения на поставените за решаване проблеми и казуси в изискванията.Избор на решение и технически подход за реализация.Активно участие в процеса на оценка на работата по имплементиране и предоставяне на окончателната оценка на Ръководителя на проекта/екипа, преди да започне процеса по имплементация. Ако някои задачи не могат да бъдат оценени в разумно кратко време, дава оценка на времето, необходимо за изследване, анализ или прототипиране на решението, така че на следващ етап да може да бъде оценено. Отговаря впоследствие за изпълнението на задачите в срок.Участва активно във фазата на имплементация, като:<ul style="list-style-type: none">Участва в разпределянето на задачите;Реализира част от задачите;Отговаря за решаването на техническите проблеми на всички програмисти в екипа. Пряко свързано е със спазването на сроковете за изпълнение на задачите на другите програмисти. Преценява до каква степен и как да включи други програмисти от екипа за

Handwritten mark

Handwritten scribble

Handwritten signature

Роля	Отговорности
	<p>решаването на техническите проблеми или да поиска външна помощ;</p> <ul style="list-style-type: none"> ○ Отговаря за координацията на работата м/у програмистите в екипа; ○ Отговаря за периодичното провеждане на прегледи (ревюта), включващи инспекция на кода и проверка на функционалността по отношение на това как са имплементирани изискванията, т.е. дали отговарят на базови технически критерии и функционални изисквания. Ревютата следва да се провеждат с целия екип, преди да се дадат задачите за тестване от тестера. В резултат от ревюто, задачите се приемат от Ръководителя за <i>тестване</i> или се връщат към програмистите за доработка; <ul style="list-style-type: none"> ● Подпомага Тестера да си изпълни ефективно своите задължения по тестване, като: <ul style="list-style-type: none"> ○ му помага да разбере как даденото изискване е реализирано или ○ като му предоставя скрипт или тестов код за генериране на данни или тестова функционалност, нужна за тестване на приложението; ● Отговаря за деплойването на приложението в тестова и продукционна среда; ● Отговаря за техническото обучение и навлизането на нови колеги в проекта.
<p>Експерт по Контрол и осигуряване на качеството (Ключов експерт)</p>	<ul style="list-style-type: none"> ● Отговаря за всички дейности по осигуряване и контрол на качеството, включително планиране на качеството, разработване на тестови сценарии и изпълнение на тестове; ● Разработва план за осигуряване на качеството в проекта, включващ мерки и процеси за осигуряване и контрол на качеството, които ще бъдат прилагани; ● Участва в извършването на контрол на качеството на разработваните програмни продукти чрез: <ul style="list-style-type: none"> ○ Разработва тестови сценарии; ○ Изпълнява тестови сценарии; ○ Разработва и прилага автоматизирани тестове; ○ Извършва вътрешни тестове на разработения софтуер, включително и автоматизирани тестове; ○ Координира работата на тестерите; ● Осигуряване на качеството на процесите и тяхното непрекъснато подобряване; ● Участва в изпълнението на приемни тестове.
<p>Програмист</p>	<p>Програмиране - създаване на софтуерни продукти или компоненти и модули от тях с приетите методи и средства; Спазване и прилагане на приетата технология за създаване на софтуер; Отговорност за работоспособността и ефективността на създадения софтуер;</p>

Роля	Отговорности
Специалист по внедряване и поддръжка на софтуер	<p data-bbox="467 202 1146 265">Първично тестване на модулите, върху които работи; Участва в код ревю.</p> <p data-bbox="467 271 852 303">По отношение на внедряване:</p> <ul data-bbox="467 340 1475 1187" style="list-style-type: none">• Разработване на план за внедряване;• Инсталиране и конфигуриране на базов софтуер;• Инсталиране и конфигуриране на специализиран разработен софтуер;• Параметризиране на софтуерни системи: настройки, работни процеси, шаблони за справки и др.• Инициализиране на софтуерни системи: въвеждане/зареждане на начални данни и номенклатури, настройване на права и роли и въвеждане на потребители и др.• Мигриране на данни от стари системи: анализ на съществуващите данни, планиране на миграцията, разработване на инструменти за мигриране на данните, зареждане и проверка на данните в новата система;• Провеждане на обучения на администратори, ключови потребители и потребители;• Подпомагане на потребителите при започване на работа със системата;• Следене и оптимизиране на системата при започване на експлоатация;• Планира провеждането на обучения за представители на Възложителя;• Подготвя провеждането на обучения – учебни материали, учебни казуси, тестове и т.н.;• Организира провеждането на обучения;• Участва в провеждането на обучения;• Администрира процеса по провеждане на обучения. <p data-bbox="467 1237 871 1269">По отношение на поддръжката:</p> <ul data-bbox="467 1306 1475 1701" style="list-style-type: none">• Осъществява техническа поддръжка на приложен софтуер;• Осъществява техническа поддръжка на базов софтуер;• Подпомага работата на потребителите;• Регистрира работата по поддръжката в специализираните инструменти, включително прави описания на разрешените проблеми;• Следи за изпълнението на споразуменията за ниво на поддръжка за системите, за които отговаря;• Осъществява мониторинг на системите, които поддържа;• Създаване и изпълнение на процедури за резервиране на данни и системи;• Приложно и системно администриране.
Тестер	<p data-bbox="467 1711 1439 1774">Осъществява контрол на качеството на разработваните програмни продукти чрез:</p> <ul data-bbox="467 1780 1412 1949" style="list-style-type: none">• Разработва тестови сценарии;• Изпълнява тестови сценарии;• Разработва и прилага автоматизирани тестове;• Извършва вътрешни тестове на разработения софтуер, включително и автоматизирани тестове;

Роля	Отговорности
	<ul style="list-style-type: none"> Участва в приемните тестове; Документира резултатите от изпълнените тестове.

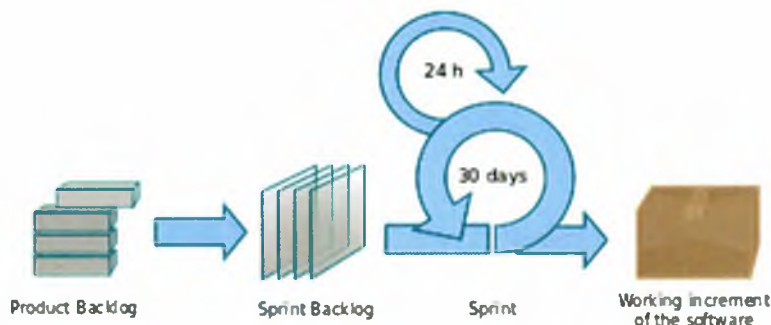
3.1.5. Участие на експертите в дейностите по проекта

№	Етап/Дейност	Подетап	Участващи експерти
1	Дейност 1: Доставка, адаптиране и внедряване на ИИС за инспекция на кораби		
1.1	Етап 1: Анализ и проектиране	Анализ на данните и изискванията	Бизнес анализатор Системен архитект
		Изготвяне на Спецификация на софтуерните изисквания	Бизнес анализатор Системен архитект Експерт по Контрол и осигуряване на качеството
1.2	Етап 2: Конфигуриране и адаптиране	Конфигуриране на софтуерното решение	Бизнес анализатор Системен архитект Експерт по Контрол и осигуряване на качеството Програмист Специалист по внедряване и поддръжка на софтуер Тестер
		Адаптиране	Експерт по Контрол и осигуряване на качеството Програмист Тестер
		Тестване	Експерт по Контрол и осигуряване на качеството Програмист Тестер
2	Дейност 2: Доставка, инсталиране и конфигуриране на ИКТ инфраструктура за ИИС за инспекция на кораби		
2.1	Етап 3: Доставка и внедряване	Доставка и инсталиране	Специалист инфраструктурни решения
		Обучение	Бизнес анализатор Специалист по внедряване и поддръжка на софтуер
		Внедряване и съпровождане	Специалист по внедряване и поддръжка на софтуер Програмист Тестер

Ръководителят на проекта участва във всички дейности, етапи и задачи в проекта.

3.1.6. Координация в екипа на Изпълнителя

Координацията на екипа ще бъде реализирана чрез прилагането на Scrum. Scrum е итеративна, инкрементална рамка за управление на проекти.



Представители на Възложителя могат да участват в планиращите и заключителните срещи на всеки спринт.

Подробно описание на подхода за екипна координация е представено по-долу в т. 5.3.3.

3.2. Управление на комуникацията

3.2.1. Проектна комуникация

Управлението на комуникацията в проекта включва процесите, необходими за осигуряване на навременно и точно генериране, събиране, разпространение, съхранение и структуриране на проектната информация. Проектните ръководители прекарват значителна част от времето си в общуване с членовете на екипа и заинтересованите лица – външни или вътрешни за организацията.

Ефективната съвместна работа на проектната организация налага въвеждането и съблюдаването на набор от формални правила за комуникация. Тези правила обикновено приемат формата на т.нар. *План за комуникация*, който зависи от големината, структурата, капацитета и зрелостта на проектната организация. Затова, правилата за комуникация в проекта следва да бъдат съгласувани между двете страни в стартовата фаза на проекта, едновременно със създаването на проектната организация. Нашият дългогодишен опит в реализацията на ИТ проекти с разнообразна сложност, големина и продължителност, показва, че като минимум, планът за комуникация трябва да включват всички аспекти на формалната комуникация: правила, канали и инструменти за обмен и контрол на информацията, подготовка и управление на документацията, организиране и провеждане на срещи. В духа на добрите практики при управлението на проекти, ние предлагаме спазването на посочените по-долу минимални препоръчителни правила.

3.2.2. Правила и канали за комуникация

За да бъде гарантирано успешното изпълнение на проекта при ефективно сътрудничество между всички членове на проектната организация, следните правила за формална комуникация е препоръчително да бъдат стриктно съблюдавани:

Определяне на език на комуникация: Езикът за комуникация в проекта, между екипите на Възложителя и Изпълнителя, е български.

Дефиниране на канали за комуникация: Например, традиционна поща и/или куриерска служба, избрана и одобрена от двете страни, електронна поща, мобилни и/или фиксирани телефони.

При комуникация чрез традиционна поща или предаване на документация на ръка в деловодство, всеки документ следва да носи уникален изходящ номер и дата на изпращача и уникален входящ номер и дата на получателя;

При комуникация чрез електронна поща всяко съобщение следва да съдържа: в полето „До: (To:)“ електронния адрес на всеки получател, от когото се очаква отговор; в полето „Копие до: (cc:)“: електронния адрес на всеки получател, за сведение на когото е съобщението; в полето „Относно (Subject:)“: темата на съобщението, на български език; в полето „Съдържание (Body:)“: самото съобщение, както и името и контактите на изпращачия го за улесняване на обратната връзка. Препоръчва се активирането на функцията „Връщане на потвърждение за прочитането (Read Receipt)“. Отговор трябва да бъде изпратен в рамките на 2 работни дни, а ако това не е възможно трябва да бъде отговорено със срока, в който се очаква да има отговор. При последваща комуникация по същата тема следва да се отговаря чрез даденото съобщение чрез функцията „Отговори на всички (Reply-to-all)“. В допълнение, страните могат да съгласуват и позволения формат и кодиране на електронните съобщения, например HTML или Plain Text и Cyrillic KOI8-R. В зависимост от зрелостта, капацитета и нуждите на проектната организация могат да бъдат одобрени и правила за комуникация чрез електронно подписани с удостоверения за универсален електронен подпис (УЕП) съобщения.

При комуникация чрез мобилни и фиксирани телефони е препоръчително първо да се използват мобилните номера (след съгласуване), като при пропуснато повикване търсеното лице има ангажимент да върне обаждането в рамките на работния ден. Добра практика е да се активира Call Waiting функцията чрез мобилните апарати.

Дефиниране на правила за работа с документи, например:

Работни документи се предават на адреса на електронната поща на съответните получатели, включително до Ръководителя(ите) на екипа(ите), на които получателите са членове.

Официални документи се предават в деловодството на Възложителя, съответно на рецепцията на адреса за кореспонденция на Изпълнителя, чрез традиционна поща и куриерска служба, избрана и одобрена от двете страни, или на ръка от представител на съответната страна-подател.

При поискване на коя да е от страните може да бъде осигурен работен документ в хартиен вид (не по ел. поща). Времето за отговор по работен документ е 2 работни дни. Времето за експертен отговор на официален документ, с изключение на работните документи, които се предават в писмен вид, е 5 работни дни. Изпратен документ след 17 часа се води за изпратен на следващия работен ден.

Всички официални отговори на официални документи, както и искания за промени в изискванията трябва да стават в писмен вид – хартиени документи изпратени чрез традиционна поща или електронни документи подписани с Универсален електронен подпис и изпратени чрез електронна поща.

Всички отговори на работни документи, както и искания за срещи трябва да стават чрез електронна поща.



3.2.3. Управление на документацията и инструменти за комуникация

В допълнение към предложеното, страните е препоръчително да съгласуват и правила за подготовка и управление на документацията, които в процеса на изпълнение на проекта да бъдат стриктно съблюдавани:

Определяне на език на документацията: Работните и официални документи се съставят и поддържат на български език.

В отделни случаи, например при използването на специфична техническа литература, Ръководителите на проекта могат да се договорят конкретните документи да бъдат предоставени и само на английски език или друг работен език.


Поддържането на даден документ на два езика едновременно е по допълнителна предварителна договорка.

Определяне на лицата, които работят с даден документ, например:

За всеки документ се указва неговият собственик. Единствено собственикът на документа има право да съгласува и одобрява промени в документа.

Списъкът с получателите на даден документ се определя от собственика му, вписва се в документа и се довежда до знанието на Ръководителите на проекта.

За всеки документ се указват: името на проекта, името на документа, текущата версия и дата, история на промени, реквизити на съставилия и одобрилия го, съдържание, номерация на страниците и общ брой страници.



Добра практика е да бъде поддържан регистър на документите, в който се описват и актуализират всички документи със своите атрибути, като: наименование на документа, наименование и адрес на файла на документа в електронен вид, собственик на документа, версия и статус на документа.

Инструментите, използвани за целите на този проект, следва да бъдат съгласувани и одобрени в стартовата фаза на проекта. Тук сме предложили препоръчителен набор от популярни инструменти, които могат да бъдат заменени с други версии или с техни еквиваленти, според наличността на лицензи при Възложителя, удобството и компетентността на участниците в проекта:

Стандартен уеб-браузър, например Internet Explorer, версия 9.0 или по-висока;

Microsoft Office Outlook: ще бъде използван за обмяна на електронни съобщения, изпращане и получаване на покани за срещи, контакти и т.н.


Microsoft Office Project: ще бъде използван за проследяване на прогреса на проекта във времето чрез визуализиране на Проектния план.

Microsoft Office Word: ще бъде използван за обмяна на работните версии на всички документи по проекта (във формат .doc);

Microsoft Office Excel: ще бъде използван за обмяна и редакция на електронни таблици (във формат .xls);

Microsoft Office PowerPoint: ще бъде използван за обмяна на презентации;

Microsoft Office Visio: ще бъде използван за обмяна и редакция на схеми, диаграми и т.н.



Microsoft SharePoint Portal: ще бъде използван по време на проекта за съхраняване на документи и колективна работа.

3.2.4. Провеждане на срещи

Постигането на целите и резултатите на проекта налага в различна степен организирането на технически (работни), управленски (проектни) и стратегически (бизнес) срещи. Препоръчително е срещи на участниците в проекта на всички нива да се провеждат периодично. С цел постигане на резултатност и ефективност при организирането и провеждането на срещи трябва да се следват някои формални правила. Правилата за организиране и провеждане на срещи се съгласуват между двете страни в стартовата фаза на проекта. Препоръчително е спазването на следните минимални изисквания:

Определяне на необходимия срок на предизвестие при свикване на среща, например: Срещите за следващата седмица се планират не по-късно от 16 часа в четвъртък на настоящата седмица или срещите се организират с предизвестие не по-малко от 24 часа.

Предварително обявяване на програмата: Темите на всяка среща се дефинират и обявяват предварително от организатора на срещата, като се включва и преглед на темите и протокола от предишната среща.

Предварително осигуряване на документация: Всички документи, например протоколи, техническа или проектна документация, които са нужни за срещата, се предоставят не по-късно от обяд на деня предхождащ деня на срещата.

Подготовка на материали: Всички материали, необходими по време на срещата, например проектор и/или флипчарт, се осигуряват от домакина на срещата.

Определяне на език: Работният език на срещите е български.

Протоколиране на срещите: Организаторът на срещата има отговорността за нейното протоколиране (в случаите когато това не е описано в ТЗ), като той може да делегира отговорността за изготвяне на протокола чрез съгласуване с участниците и анонсиране на протоколчика в началото на всяка една среща.

Определяне на срок за изготвяне и разпращане на протокол от среща: Протоколи от срещите се подготвят и изпращат на всички участници не по-късно от два работни дни след приключване на срещата по предварително съгласуваните канали за комуникация (виж по-горе).

Съгласуване на протокол от среща: Ако до два работни дни след получаване на протокола от срещата участниците не уведомят организатора на срещата в писмен вид (в съответствие с одобрените канали за комуникация) за своите забележки към него, протоколът се счита за одобрен.

Аудио протоколи: Организаторът на срещата има правото да поиска използването на аудио протоколиране, съгласувано с участниците в срещата. В случай на съгласие, организаторът има отговорност да осигури необходимите технически средства, както и да разпространи протокола сред участниците в предварително уговорен формат (виж също инструменти за комуникация по-горе) и при спазването на всички останали правила за комуникация. Следвайки добрите практики, препоръчително е в случаите на аудио протоколиране да бъде изготвено резюме в писмен вид.

Свикване на срещи по спешност: Срещи извън посочения график могат да бъдат организирани при необходимост. Заявка за такава среща трябва да бъде изпратена от организатора ѝ, като се препоръчва толеранс от поне един ден за да се даде време на всички участници да се подготвят.

Препоръчително е срещи на Техническите ръководители или Ръководителите на екипи от двете страни да се провеждат ежеседмично. Основната цел на тези срещи е да се дискутират и съгласуват оперативни въпроси, които изискват координирани експертни действия на страните.

Препоръчително е срещи на Ръководителите на проекта от двете страни да се провеждат не по-рядко от веднъж месечно. Основната цел на тези срещи е да се дискутират въпроси, влияещи на обхвата, сроковете и ресурсите на проекта.

Препоръчително е срещи на Комитета за управление на проекта да се провеждат не по-рядко от веднъж на два месеца. Основната цел на тези срещи е да се дискутират стратегически въпроси, важни за успешното приключване на проекта, като съществени промени в целите и очакваните резултати от проекта, управление на рисковете и цялостен контрол на текущото състояние на изпълнение на проекта.

3.2.5. Канали за комуникация

Тип на комуникацията	Комуникационен канал	Страни в комуникацията
Официална (формална)	Писма, входящи в деловодството на организацията	Между проектните ръководители; Между ръководителите на организацията.
	Факс	
	Работни срещи, протоколирани с протокол	
Оперативна	Електронна поща	Между членовете от екипа, отговорни за темата на комуникацията.
	Телефон и др. средства за провеждане на разговори (Lync, Skype и т.н.)	
	Факс	
	Работни срещи	
	Портал за поддръжка	

3.2.6. Докладване

В хода на изпълнение на договора Изпълнителят ще изготви и предостави на Възложителя:

- Встъпителен доклад - Встъпителният доклад ще бъде предоставен в двуседмичен срок от подписването на договора и да съдържа описание минимум на:
 - Подробен работен план и актуализиран времеви график за периода на проекта;
 - Начини на комуникация;
 - Отговорни лица и екипи.

Встъпителният доклад следва да бъде одобрен от Възложителя.

- MA*
- Междинен доклад - Междинните доклади ще бъдат представяни и да се предават при приключване на всяка от дейностите и поддейностите и/или при настъпване на събитие. Междинните доклади ще съдържат информация относно изпълнението на дейностите и поддейностите по предварително изготвения проектен план. Като минимум междинните доклади ще съдържат следната информация:
 - Общ прогрес по дейностите през периода;
 - Постигнати проектни резултати за периода;
 - Срещнати проблеми, причини и мерки, предприети за преодоляването им;
 - Рискове за изпълнение на свързани дейности и на проекта като цяло и предприети мерки;
 - Актуализиран план за изпълнение, ако има такъв.

Всеки междинен доклад следва да бъде одобрен от Възложителя.

- Окончателен доклад - В края на периода за изпълнение ще бъде предоставен окончателен доклад. Окончателният доклад трябва да съдържа описание на изпълнението и резултати.

Докладите ще се изпращат до отговорния служител на Възложителя. Всички доклади се представят на български език в електронен формат и на хартиен носител. Докладите се одобряват от отговорния/отговорните служител/служители в срок до 5 работни дни.

Всички доклади ще бъдат представени на Възложителя на български език на хартиен и на електронен носител. Представянето на докладите трябва да се извършва чрез подписване на двустранни предавателно-приемателни протоколи, подписани от представители на Изпълнителя и на Възложителя.

Примерно съдържание на междинни доклади:

1. Въведение

2. Общ преглед на състоянието на проекта

Общ статус:

[Цвят]

[Резюме]

3. Прогрес по конкретните дейности

3.1. Дейност X

3.1.1. Общ прогрес по дейността

3.1.2. Постигнати проектни резултати

3.1.4. Планирани дейности в следващия период

4. Срещнати проблеми и отворени въпроси

№	Описание	Предложение за решение	Решение/ коментар

5. Рискове

Описват се във формата на регистър на рисковете, представен в плана за управление на риска.

6. Препоръки за текущото състояние на проекта

7. Актуализиран план за изпълнение на проекта



3.3. Управление на качеството

3.3.1. Стратегия за качество в проекта

Стратегията за осигуряване на качеството определя дейностите, които ще осигурят, че контролът на качеството се прилага ефективно и механизмите за контрол на качеството са адекватно планирани, графици са спазвани, преглеждани и актуализирани.

Осигуряване на качеството (QA) се отнася до систематично измерване, сравнение със стандарти, мониторинг на процесите и свързаните с тях дейности за осъществяване на обратна връзка, която спомага за предотвратяването на грешки. Това контрастира с процеса по контрол на качеството, който се фокусира върху изхода на процесите.


Осигуряването на качеството се основава на два базови принципа:

- 1) продукта трябва да може да изпълнява предназначението си;
- 2) откритите грешки трябва да се отстраняват възможно най-бързо.

Осигуряването на качеството включва управление на качеството на всички базови продукти, компоненти на системата и процеси свързани с изграждането, включително тези на управление и мониторинг и контрол.

Критериите за определяне на качеството се определя от заинтересованите лица и основно потребителите на системата. Те са свързани и с разходите за прилагане на необходимите изисквания, които зависят както от наличните бюджети, потребителите, така и от самия продукт, тъй като определено ниво на качеството е свойствено за точно определени категории продукти.


Качеството на продукта може да бъде разглеждано от два аспекта:

- 
- 1) Какви характеристики притежава;
 - 2) До каква степен тези характеристики са реализирани.

Притежаваните характеристики се задават при изготвянето на функционалната спецификация на проекта. Притежаваните характеристики са ограничени от обхвата на договора, от разполагаемото време и от наличните ресурси. Предмет на тестването е да установи до каква степен тези характеристики са осъществени. Поради това то се изпълнява след завършване на разработката. При процес на бърза разработка, то се изпълнява паралелно с разработката, в края на всяка итерация, но е възможно да се окаже, че изисква значително повече ресурси спрямо посветеното тестване, в края на проекта. При този похват обаче се спестява време, тъй като голяма част от дейността се изпълнява паралелно на разработката. Частта от дейността, която не може да се изпълни паралелно е осъществяването на приемните тестове. Приемните тестове се осъществяват на базата на предварително изготвени и съгласувани с Възложителя приемни сценарии. Те имат за цел да покажат, че системата може да бъде въведена в експлоатация.

Като част от проектния план ще бъде разработен План за осигуряване на качеството. След като Планът за осигуряване на качеството бъде разработен и приет от Възложителя, той ще бъде водещият документ за организиране на процесите по управление на качеството в проекта.

Всеки етап в жизнения цикъл на проекта ще се отрази като артефакт, даващ основа за следващия етап. Чрез контрола на качеството тези артефакти ще бъдат проверявани преди да са използвани в следваща фаза, което ще гарантира верността й. Те също ще бъдат верифицирани съобразно техническата спецификация.



Всеки документ и материал, който ще бъде доставян на Възложителя или други външни лица подлежи на контрол на качеството.

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

- Документиране на Системата в изходния код, минимум на ниво процедура/функция/клас;
- Покритие на минимум 50% от изходния код с функционални тестове [в случай на надграждане на съществуваща система – 50% от новата функционалност и 20% от съществуващата];
- Използване на continuous integration практики;
- Използване на dependency management.

3.3.2. Процедури за контрол на качеството

Съществуват няколко метода, които ще бъдат използвани при осигуряването на проекта и материалите по него в съответствие с подходящите стандарти за качество:

- Вътрешни прегледи на проекта - Това са работни сесии на екипа по проекта, в които екипът преглежда всички материали по конкретна фаза преди да определи методологията за официален преглед. Прегледът се извършва от ръководителя на проекта и ръководителя на разработката.
- Инспекции - Това е съвместен преглед на материали от представители на Възложителя и изпълняващия екип за целите на проверка и приемане.

3.3.3. Роли и отговорности в процесите по осигуряване и контрол на качеството

Ръководителят на екипа ще бъде отговорен за:



- Определяне на стратегия за осигуряване на качеството;
- Установяване на процедури за осигуряване на качеството;
- Определяне на задачи и отговорности на членовете на екипа свързани с изпълнението на процедурите по осигуряване на качеството;
- Наблюдение на процесите по осигуряване на качеството и извършване на коригиращи действия, при необходимост.

Експерт по контрол на качеството:

- Участва в определянето на стратегия и процедури за осигуряване на качеството;
- Организира изпълнението на процедурите и стратегията за осигуряване на качеството;
- Разработва подробен тестов план.

Всеки ръководител на технически екип:

- Ще бъде отговорен за спазването на установените в проекта стандарти и процедури за осигуряване на качеството в екипа, който ръководи;
- Ще сигнализира на Ръководителя на проекта за потенциални слабости в качеството на разработвания в екипа и другите екипи продукт и ще дава мотивирани предложения за подобряване на стандартите за качество.



Всеки член на екипа:

- Ще бъде отговорен за прилагането на стандартите за качество на проекта при изпълнението на своята работа;
- Ще изпълнява възложените му задачи за осигуряване на качеството;
- Ще сигнализира за дефекти и грешки, намерени при изпълнение на собствените си задачи, включително когато не са пряко свързани с текущата задача.


Представители на клиента ще имат възможност да вземат участие в процесите за осигуряване на качеството.

Ръководителят на проекта ще бъде отговорен за общото качество на проекта от страна на ИЗПЪЛНИТЕЛЯТ.

Ръководител на проекта ще бъде отговорен за:

- Определяне на стратегия за осигуряване на качеството;
- Установяване на процедури за осигуряване на качеството;
- Определяне на задачи и отговорности на членовете на екипа свързани с изпълнението на процедурите по осигуряване на качеството;
- Наблюдение на процедурите по осигуряване на качеството и извършване на коригиращи действия, при необходимост.

Експерт по контрол на качеството:

- 
- Участва в определянето на стратегия и процедури за осигуряване на качеството;
 - Организира изпълнението на процедурите и стратегията за осигуряване на качеството;
 - Разработва подробен тестов план.

Всеки ръководител на технически екип:

- Ще бъде отговорен за спазването на установените в проекта стандарти и процедури за осигуряване на качеството в екипа, който ръководи;
- Ще сигнализира на Ръководителя на проекта за потенциални слабости в качеството на разработвания в екипа и другите екипи продукт и ще дава мотивирани предложения за подобряване на стандартите за качество.

Всеки член на екипа:

- Ще бъде отговорен за прилагането на стандартите за качество на проекта при изпълнението на своята работа;
- Ще изпълнява възложените му задачи за осигуряване на качеството;
- Ще сигнализира за дефекти и грешки, намерени при изпълнение на собствените си задачи, включително когато не са пряко свързани с текущата задача.

Представители на клиента ще имат възможност да вземат участие в процесите за осигуряване на качеството.

3.4. Мерки за мониторинг и контрол

Наблюдението и контролът на работата по проекта е процесът на проследяване, преглед и регулиране на напредъка с цел покриването на целите за производителност, дефинирани в плана за управление на проекта. Този процес е част от групата за управление на интеграцията в проекта.

Предлагаме механизми за проследимост и отчетност на изпълнението, които да осигурят ефективно управление на проекта. Ще бъде проследявана степента на постигане на целите при съпоставяне на действителните и очакваните резултати от изпълнението.

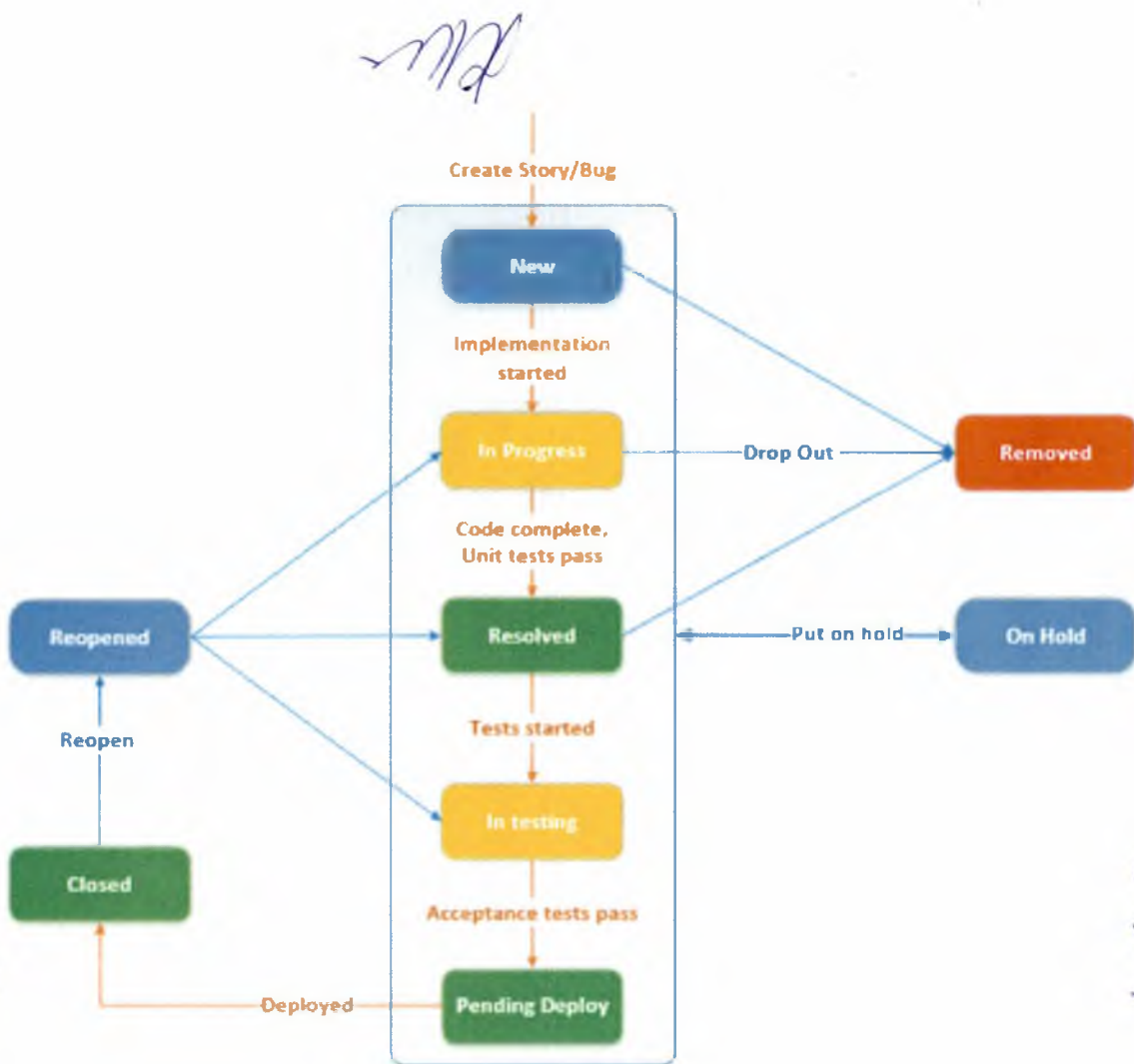
Във фазата на планиране ще бъдат разработени следните артефакти, които ще бъдат основа за бъдещия мониторинг:

- Детайлен план за изпълнение;
- Ключови индикатори на начини за измерването им;
- Бюджет на проекта.

Най-значителният ресурс, който ще бъде вложен в изпълнението на поръчката е времето на проектните екипи. Тяхната работа ще бъде планирана, разпределяна и регистрирана чрез Atlassian Jira – инструментална среда за управление на съвместната работа на софтуерни екипи.

За управлението на задачите (work items) ще се използват следния процес и статуси:

000149



В инструменталната среда ще се въвежда и съхраняват данни за планирано и реално изработено време за всеки индивид в проектния екип.

За анализ на разходваните средства и оценка на постигнатите резултати предлагаме да се използва методът на спечелената стойност (Earned value management / EVM). EVM е техника от дисциплината управление на проекта, която служи за измерване на напредъка на изпълнението и прогреса по обективен начин. Тази техника комбинира в едно измерването на напредъка относно обхвата, графиката и разходите в една интегрирана система за оценка. Методът дава възможност за изготвяне на акуратна оценка на проблемите с производителността на проекта.

Спечелената стойност се дефинира като обема свършена от началото на проекта до момента запланувана работа или:

$$EV = \sum_{Start}^{Current} PV(Completed)$$

Цели: Анализът на спечелената стойност е най-често използваният начин за измерване на напредъка. Използва се за ревизия на планираните график и бюджет за оставащата част от проекта и/или за да се идентифицира нуждата от промяна на обхвата на проекта.

Начин на измерване: За да бъде осъществена EVM калкулацията е необходимо наличието на следните предусловия:

- Необходим е пълен списък със задачи;
- Необходим е списък на ресурсите с техните часови ставки;
- Необходим е списък на всички други разходи;

- Необходим е одобрен план-график;
- Необходим е текущ статус на проекта;
- Необходима е дата на изготвяне на доклада.

При измерването се използват следните индикатори:

- PV или често използвана като BCWS – Planed value/Планирана стойност, и е еквивалентна на Budgeted Cost of Work Scheduled/Бюджетна стойност на планираната работа;
- EV или често използвана като BCWP – Earn value/Спечелена стойност, и е еквивалента на Budgeted Cost of Work Planned/Бюджетна стойност на планираната работа;
- AC или често използвана като ACWP – Actual Cost/Реални разходи, и е еквивалентна на Actual Cost of Work Planned/Реални разходи за извършване на планираната работа;
- SV – Schedule variance/Разминаване в графика;
- CV – Cost variance/Разминаване в бюджета;
- EAC – Estimate at completion/Прогноза към датата на приключване;
- BAC – Baseline at completion/Планирана стойност към датата на приключване;
- ETC – Estimate to complete/Прогнозата за завършване на оставащата работа по проекта.

Използват се методи за оценка базирани на Изменението в графика и Изменението в разходите. Това става по следните формули:

- Изменение спрямо графика - Метода служи за управление на графика. Той не включва в себе си акумулиране на реалната стойност (AC), а се базира на оценка на прогреса спрямо времевия план. Метода се изпълнява самостоятелно в типичните малки и средни по големина проекти.

$$SV = EV - PV$$

Оценката става на следния принцип: SV по-големи от 0 показват, че проекта изпреварва графика. Разминаването в графика ще бъде 0 в края на проекта, тъй като всички планирани стойности са били „спечелени“.

- Използва се и така наречения Schedule Performance Index (Индекс за производителността спрямо графика). Той се изчислява съобразно формулата:

$$SPI = \frac{EV}{PV}$$

SPI > 1 се смята за много добър, тъй като тогава проекта изпреварва графика.

- Измерване спрямо разходите – Метода служи за управление на разходите. Измерва се общата планирана стойност (BCWS) на края на проекта. Ако в бюджета има резерви за управление те типично не се включват в оценката. Прилагат се следните формули:

$$CV = EV - AC$$

CV > 0 се считат за добри, тъй като проекта икономисва средства.

- Използва се и така наречения Cost Performance Index (Индекс за производителността спрямо бюджета). Той се изчислява съобразно формулата:

$$CPI = \frac{EV}{AC}$$

CPI > 1 са добри, тъй като проекта икономисва средства;

CPI = 1 означава, че проекта изразходва средства съобразно плана;

CPI < 1 означава, че проекта изразходва средства над планираните.

Твърде високи стойности на CPI (в някои случаи и 1.2 е високо) може да означава, че плана е твърде консервативен и производителността се измерва спрямо лош първоначален план.

Начин на отчитане и проследимост на извършената работа: Изготвянето на доклад за статуса на проекта посредством EVM е напълно автоматизирано с използването на предложението от нас софтуер за управление на проекта, а именно Microsoft Project 2010. Продукта поддържа и възможност за избор между това дали калкулацията на Индекса на бюджетните разходи на извършената работа (BCWP) да е базирана на % Complete или на Physical % Complete. Те кореспондират на напредъка спрямо графика и напредъка спрямо извършената работа. Всички от гореизброените стойности могат да бъдат автоматично изчислени на база на въведената информация. Възложителя ще има достъп до системата, а също така ще получава извадка за състоянието с регулярните отчети по договора.

Предлагаме да бъдат приложени следните видове мерки за контрол, относими към цялостния процес на изпълнение на проектните дейности, които ще прилага с цел качествено и навременно изпълнение на поръчката:

- Контрол на изпълнението на графика – чрез сравнение между базовия график и текущия напредък. Ще се използват Schedule Performance Index (SPI) и Schedule Variance (SV).
- Контрол на изразходване на бюджета – чрез сравнение между базовия бюджет и реално вложените разходи и време. Ще се използват Cost Performance Index (CPI) и Cost Variance (CV).
- Контрол на качеството, съгласно описаното в предложението.

4. Етапи и дейности в обхвата на поръчката

4.1. Описание на дейностите

4.2. Резултати от изпълнението по етапи и дейности

No	Етап/Дейност	Срок	Резултати
1	Анализ и проектиране		
	<p>Етапът обхваща изпълнението на следните задачи:</p> <ul style="list-style-type: none"> • Детайлно проучване на изискванията към новите функционалности на системата; • Дефиниране на детайлните изисквания и бизнес процеси, които трябва да се реализират в системата; • Дизайн на системата, хардуерната и комуникационната инфраструктура; • Проектиране на необходимите промени и доработки в модули и функционалности, както и тяхната конфигурация, въз основа на анализа и изискванията; • Проектиране и специфициране на интеграционен интерфейс с румънския компонент на ИИС и модел на обменяните данни. 	<p>28.09.2018 г</p>	<ul style="list-style-type: none"> • Спецификация на софтуерните изисквания • Системната архитектура • Спецификацията на интеграционен интерфейс с румънския компонент



000153


000154

	<ul style="list-style-type: none">• Изготвяне на план за техническа реализация;• Определяне на потребителския интерфейс;• Изготвяне на подробна Спецификация на софтуерните изисквания;• Изготвяне на Системна архитектура;• Изготвяне на Спецификация на интеграционен интерфейс с румънския компонент на ИИС.		
2	Конфигуриране и адаптиране		
	<p>Етапът на конфигуриране и адаптиране включва изпълнението на следните задачи:</p> <ul style="list-style-type: none">• Конфигуриране на модулите на системата съгласно изискванията на техническото задание и техническата спецификация;• Актуализиране на спецификациите, разработени в рамките на Етап 1• Подготвяне на тестови случаи за приемане на информационната система, съобразени с одобрената от Възложителя	<p>29.03.2019 г.</p>	<ul style="list-style-type: none">• Инструкции за инсталация и скриптове за създаване на базата данни, включващи:<ul style="list-style-type: none">○ Детайлно описание на базата данни;○ Описание на софтуерните модули.• Изходен код на разработените в обхвата на поръчката модули и функционалности;• Документация на системата, вкл.:<ul style="list-style-type: none">○ Ръководство за потребителя;○ Ръководство за администратора;• План за тестване;• Тестови сценарии за приемане на системата.

	<p>спецификации на изискванията, включващи:</p> <ul style="list-style-type: none"> ○ Функционални тестове; ○ Интеграционни тестове. <ul style="list-style-type: none"> • Провеждане на вътрешни тестове на системата (в тестова среда на разработчика); • Създаване специализирани изгледи на данните за справочни цели, инсталиране и конфигуриране на справочната система; • Конфигуриране на примерни справки. • Разработване на документация за потребителите и администраторите на системата; • Изготвяне на детайлни тестови сценарии за приемане на системата 		
3	Доставка и внедряване		
	<p>В този етап Изпълнителят ще осигури експлоатационната среда на информационната система като ще изпълни следните задачи:</p> <ul style="list-style-type: none"> • Инсталиране, конфигуриране и настройка на програмните 	<p>30.04.2019 г.</p>	<ul style="list-style-type: none"> • Актуализирана системна документация (в случай на необходимост); • План за обучения; • Учебни материали; • Протоколи за проведени обучения; • Протокол за успешно проведени приемни тестове; • Протоколи за доставка, инсталация и конфигурация на ИКТ инфраструктурата по Дейност 2;

000155

	<p>компоненти на системата в условията на експлоатационната среда на Възложителя.</p> <ul style="list-style-type: none">• Въвеждане на всички номенклатури, класификатори и конфигурации (ако е необходимо въвеждането на такива)• Въвеждане на всички потребители, които няма да се определят с работен процес посредством системата.• Провеждане на приемателни тестове;• Доставка, инсталиране и конфигуриране на ИКТ инфраструктура за ИИС за инспекция на кораби:<ul style="list-style-type: none">○ Доставка на оборудването в локации на Възложителя○ Инсталация на сървърен шкаф○ Монтаж на UPS в сървърния шкаф○ Монтаж на блейд шаши в сървърния шкаф		<ul style="list-style-type: none">• Протоколи за доставка, инсталация и конфигурация на базов софтуер. 
--	---	--	--

- 
- | | | | |
|--|---|--|--|
| | <ul style="list-style-type: none">○ Монтаж на споделен сторидж в сървърния шкаф○ Инсталация на операционни системи○ Инсталиране и конфигуриране на 2 х Тауър сървър○ Функционални тестове на хардуерните ресурси● Актуализиране на системната документация, предадена в Етап 2;● Разработване на план за обучения;● Организиране и провеждане на обучения за следните групи и ползватели на софтуерното решение:<ul style="list-style-type: none">○ Ключови потребители – до 30 човека, разпределени в две учебни групи;○ Администратори на системата – до 5 човека. | | |
|--|---|--|--|

[Handwritten mark]



[Handwritten mark]

[Handwritten mark]

4.3. Ресурсна обезпеченост

Ресурсната обезпеченост е представена в приложения примерен план-график в *Приложение 4 Подробен график* към настоящото техническо предложение. Ролите на участниците при изпълнение на поръчката са представени в т. **3.1.4 Роли и отговорности** от настоящото приложение.

5. Описание на подход и методи за ключови дейности от изпълнението на проекта

5.1. Подход за бизнес анализ

Подходът за бизнес анализ, който Кандидатът ще прилага е представен в *Приложение 1 Предложение за технологично решение и функционалности* в т. 5 Подход за бизнес анализ към настоящото техническо предложение.

5.2. Подход за проектиране на софтуер

Методологии и добри практики:

Unified Modelling Language - UML
Business Process Modelling Notation – BPMN
SOLID

Цикъл на Деминг
Шаблони за дизайн

Средства и техники:

Microsoft Visio
Sparx Systems Enterprise Architect – софтуер за моделиране на изисквания, бизнес процеси и архитектура. Продуктът поддържа всички описани по-долу видове диаграми.

Източници:

Object Management Group – сайт и документи: www.omg.org/
www.Uml.org
<http://www.sparxsystems.com.au/>

Обосновка:

Използват се световно утвърдени процеси и принципи за проектиране, като цикъл на Деминг, SOLID, шаблони за дизайн, UML и др.
UML и BPMN са стандарти в обектно ориентираното моделиране. Enterprise Architect е утвърден софтуерен продукт, който предлага необходимата функционалност и, с който нашият екип има опит. Продуктът е популярен и сред българските разработчици и възложители.

5.2.1. Процес за проектиране

При проектирането на процесите и при разработването на ориентирана към процесите система, може да се използва и популярния цикъл на Деминг.

Важно условие за използването на цикъла на непрекъснато усъвършенстване е използването на подходящи методи за анализ на представянето на процесите.

Такива методи например могат да бъдат:

- 1) Съпоставка с „идеален“ процес. Цел: да се открият различията между действащата практика и определения като идеален вариант;
- 2) Проиграване на сценарии тип „какво ще се случи, ако...“. Цел: да се разработят няколко варианта за осъществяване на дадена дейност и да се избере най-добрия.

- 3) Съпоставка с предварително дефинирани ключови параметри на процеса. Цел: да се открият пропуски или възможности за подобрене при осъществяването на процеса.
- 4) Съпоставка между два и повече процеса в организацията. Цел: да се открият общи за организацията силни или слаби страни и да се търси начин на пренасяне на успешните практики от един процес към друг.

Използването на методи за анализ, помага на екипа натоварен с разработка и внедряване на система за управление при изготвянето на описание на процесите и на тяхното взаимодействие. Методите за анализ следва да бъдат предвидени като част от системата за управление и да се превърнат в обичайна практика в организацията.



Цикълът на Деминг

Цикълът на Деминг показва пътя към подобрения. Той стои в основата на много от съвременните концепции за развитие на бизнеса. Известен е още като "Цикъл на Шухарт", "PDCA цикъл", "PDSA цикъл" или "SDCA цикъл". Съществуват и редица други модификации на цикъла, които доказват приложимостта му в много области и ситуации.



PDCA (PDSA) цикъл

Деминг развива идеите на Шухарт, като дава по-обща названия на всеки от етапите и добавя още един етап за анализ на получената на втория етап информация.



PDCA (PDSA) цикъл

- 1) **Планиране (Plan)** - Използва се подхода 5W 1H, при който трябва да се дадат отговори на следните шест въпроса:
 - какво? (what?) – дефиниране на целите и задачите
 - защо? (why?) – обосноваване на необходимостта;
 - кой? (who?) – определяне на отговорностите;
 - как? (how?) – определяне на метода (методите) на извършване;

- 
- 
- къде? (where?) – ограничаване на полето на действие;
 - кога? (when?) – разработване на план (график).
- 2) **Изпълнение (Do)** - В началото на този етап се извършва необходимото обучение и квалификация на персонала, който ще извършва реалното изпълнение на планираното и неговото внедряване. След обучението (или паралелно с него) персоналетът изпълнява планираното и го внедрява.
- 3) **Проверка (Check)** - На този етап се проверява как е изпълнено и внедрено планираното действие и се оценяват постиженията.
- **Изучи (Study)** – използва се при PDSA цикъла - Изучаване и анализ на постигнатите резултати, с цел уточняване на по-нататъшните действия.
- 4) **Действие (Act)** - В зависимост от резултатите на проверката на този етап са възможни два вида действия:
- Въвеждане на постигнатото по-високо ниво като нов стандарт (формализиране) в организацията;
 - Извършване на коригиращи и/или превантивни действия за подобряване на постигнатото и за постигане в следващия цикъл още по-високо ниво.



Документиране



В рамките на под-процеса по Анализ ще се постави основата за реализацията и приемането на продукта. В под-процеса Документиране, се изготвя документ, който има за цел да формализира тази основа. Той трябва да бъде верифициран от Възложителя и в последствие ще бъде реализиран от екипа по разработка.

Документирането е процеса по записване на анализирания изисквания, било те функционални или технически. То се извършва паралелно на останалите процеси, но леко изостава във времето от тях. То служи като основа за следващия процес по верификация на изискванията, където възложителя трябва да потвърди коректността на извършения анализ.


При документирането първоначално се документират основните изисквания на заинтересованите лица, детайлизирани до нивото, до което те са известни. Във времето те се конкретизират и детайлизират. Поради това, процесът е итеративен и спираловиден. Преди да бъдат вписани изискванията, те трябва да бъдат SMART (виж по-горе).

Формата на документа може да бъде различен. Това може да е един документ, изброяващ всички изисквания, категоризирани по заинтересовани лица и приоритет, както и по-подробна форма, съдържаща описание, детайли и свързани документи. Този документ може да включва например следното съдържание:

- 1) Бизнес целите и целите на проекта за проследимост;
- 2) Функционалните изисквания, описващи бизнес процеса, информацията и взаимодействията с продукта или тези от тях, които са необходими;
- 3) Може да бъде текстов документ даващ списък на изискванията, моделите или и двете;
- 4) Нефункционални изисквания, като ниво на услугата, производителност, сигурност, съвместимост със стандарти и административни изисквания, възможности за поддръжка, и др.

- 
- 5) Изисквания към качеството;
 - 6) Бизнес правила;
 - 7) Въздействие върху други области на организацията, като центъра за обслужване, технически групи и т.н.;
 - 8) Въздействие върху други подразделения вътре и извън организацията;
 - 9) Изисквания към обучението;
 - 10) Изисквания към поддръжката;
 - 11) Предположения и ограничения;
 - 12) Изисквания за тестване.
- 

Техническата част на спецификацията трябва да включва различните аспекти на архитектурата на приложението, което ще бъде реализирано. Това означава, че трябва да бъдат анализирани и документирани аспекти като:

- 
- 1) Сигурност;
 - 2) Технология;
 - 3) Интерфейси;
 - 4) Миграция (Цифровизация);
 - 5) Стандарти;
 - 6) Производителност;
 - 7) Скалируемост.


Верификация

Оценка на решенията и валидиране описва начина, по който бизнес анализаторите оценяват предложените решения. За да се определи кое решение най-добре отговаря на бизнес изискванията, трябва да се идентифицират пропуските и недостатъците в предложените решения и да се определят необходимите отклонения (заобикаляния) или промени на обхвата. Тази част от анализа също описва как бизнес анализаторите оценяват разгърнатите решения, за да се види как тези решения отговарят на първоначалните изисквания, така че спонсориращата организация да може да оцени изпълнението и ефективността на предложеното решение.

При проектирането на процесите на системата за управление е важно да използваме информацията за действащите и идеалните процеси и да съдействаме на ръководството на организацията да дефинира „желаните“ процеси. Решенията за реда за протичане на дейността са изцяло от компетенциите на мениджърите на организацията и не могат, и не бива да се прехвърлят върху екипа от специалисти, които са натоварени с разработването на системата.

Процесът по верификация на изискванията следва този по документиране. Той може да бъде еднократен, т.е. да се верифицират изискванията наведнъж в една итерация или да се верифицират поетапно групи изисквания. Той има за цел проверка и формално одобрение на документираните изисквания.

Изискванията трябва да бъдат верифицирани. Документацията се представя на Възложителя и ако е необходимо се провеждат разяснителни сесии. Спецификацията задължително се документира.



5.2.2. Софтуерна архитектура

Методи за създаване на софтуерна архитектура

Софтуерната архитектура е от голямо значение при проектирането, изграждането, инсталирането и поддръжката на всяка софтуерна система. Тя определя целите, изискванията и стратегията за разработка. Архитектурата обаче има различно значение за различните участници в процеса:

- Спецификацията на системата, която ще бъде имплементирана (блупринт);
- Средство за комуникация за постигане на общо разбиране (език);
- Обосновка за избора на решения, които ще бъдат имплементирани от системата;
- Документация за сегашните и бъдещите потребители и разработчици.

Множеството гледни точки водят до необходимостта от описание на софтуерната архитектура с различни методи и похвати. Методите за описание на софтуерната архитектура могат да бъдат класифицирани като:

- Артефакт ориентирани методи: фокусът е в артефактите, които трябва да бъдат произведени с акцент върху структурното описание на системата;
- Методи на потребителските случаи: използват описанието на потребителски случаи като основен артефакт;
- Методи ориентирани към качеството: използват качествените характеристики като основен двигател за избиране на едно архитектурно решение пред друго;
- Методи от областта на приложение: използват концепции от домейна (областта на приложение), при дефинирането на архитектурата.

Независимо от избраният метод (артефакти, случаи на употреба, ориентиран към качеството или домейна), процеса на архитектурното проектиране включва следните аспекти:


- Анализ на изискванията на архитектурата, идентифициране на рискове и процес за вземане на решения;
- Документиране на характеристиките и принципите на системата и нейното развитие, за да се улесни комуникацията им със заинтересованите страни на системата;
- Оценка и сравнение на архитектурни алтернативи;
- Проверка на съответствието на имплементацията на системата с архитектурното си описание;
- Управление на промените в архитектурата

Анализ на архитектурните изисквания

Анализът на архитектурните изисквания е процес, чрез който се идентифицират целите системата и очакваните качествени характеристики. Тези изисквания трябва да бъдат пряко свързани с бизнес целите и изрично да включват заинтересованите страни.

Една от най-важните стъпки в анализа е определянето на контекста: къде завършената и доставена система ще се впише в текущата оперативна среда. Идентифицирането на заинтересованите страни, системни цели, и на сценарии и случаи на употреба зависят от правилно дефинираният обхват.

Събирането на изисквания може да се направи с помощта на различни техники, като например използването на въпросници и анкети, интервюта и анализ на съществуващата документация. Груповото събиране на изисквания може да бъде използвано за ангажиране на заинтересованите страни и да предизвика по-пълно разбиране на нуждите и насърчаване на постигането на съгласие между различните заинтересовани страни. Тези техники включват брейнсторминг, фокус групи, и работни



срещи за постигане на консенсус. Съществуват и по-официални методи, които се основават на подробен модел на събираната информация, като качествени характеристики и тяхното определение по отношение на качествените показатели.

Документиране на характеристиките

Тази дейност е в основата на процеса на разработване на архитектурата. В нея се предлагат решения от гледна точка на организация на системата и принципи за развитие.

Описание на решението трябва да бъде направено от различни гледни точки, които отразяват различни аспекти на системата като областта на приложение (домейна), функционалните изисквания, артефактите за разработката на системата, компоненти по време на изпълнение и тяхното поведение, съответствие с инфраструктурата, върху която ще се внедри решението. С цел да бъдат полезни документираните спецификации трябва да бъде едновременно достатъчно детайлни за анализ и с необходимото ниво на абстракция за разбиране.


Желателно е процеса на проектиране да приема добре дефиниран архитектурен стил с капсулиране на важните решения за архитектурни елементи и подчертаване важните ограничения на елементите и техните връзки.

Процесът обикновено е подкрепен с референтна архитектура, която включва набор от най-добри практики за домейна, на който трябва да функционира системата. В някои индустрии и технологични домейни, тази референтна архитектура е формализирана.

Анализ на архитектурни алтернативи

Целта на анализа на софтуерната архитектура е да се валидира, че изискванията за качество са били спазени в проектирането и са идентифицирани потенциалните рискове. Методите за оценка могат да се класифицират като базирани на софтуерни метрики, на сценарии или на модел на атрибутите. Методите базирани на софтуерни метрики използват идеите за ниска свързаност (зависимост) между модулите и кохезия, за да се идентифицират потенциалните проблеми с качеството. Методите базирани на сценарии се фокусират върху идентифициране на противоречиви изисквания или непълна спецификация от гледна точка на заинтересованите страни. Методите за оценка, базирани на модел на атрибутите се фокусират върху оценка на качествените показатели. Техниките за оценка могат да бъдат беседа, контролни листове, формални методи, симулация или прототипи.

Проверка на съответствието на архитектурата



Често се налага на софтуерните архитекти да удостоверят, че системата е имплементирана в съответствие с планираната архитектура. По време на изпълнението и развитието на система, понякога се налагат отклонения от определената архитектура, дори и в първата версия, поради непознаване от разработчиците, противоречащи си изисквания, късни промени в изискванията, технически трудности, къс срок за изпълнение или малък бюджет. Тези отклонения обикновено се натрупват с течение на времето, което води до феномена, известен като архитектурен ерозия.

Съответствието на архитектурата е мярка в каква степен имплементираната архитектура в изходния код съответства на планираната архитектура. За оценка на съответствието, трябва да се създаде модел на действителната система, който да се сравни с планирания архитектурен модел, за да се установят нарушения, като например непозволените зависимости.

Алтернативно, отклоненията от планираната архитектура могат да бъдат предотвратени чрез прилагане на определени шаблони за дизайн, които предлагат утвърдени решения за повтарящи се проблеми при проектирането, следващи архитектурните ограничения, използването на фреймуърци (рамки за

разработка), чиято архитектура е в съответствие с търсената архитектура на решението или с помощта на инструменти, които генерират архитектурно-съвместим код.

Управление на промените в архитектурата

Способността на една софтуерна система лесно да се адаптира към бъдещи промени в бизнес изискванията, организационния контекст, както и технологии. Тя е основна характеристика за увеличаване на икономическата стойност на софтуера и минимизиране на технологичния риск в една организация. За софтуерните системи с дълъг жизнен цикъл, е необходимо да се направи ясна оценка на способността за адаптиране към промените по време на целия жизнен цикъл на софтуера, за да се удължи продуктивният живот на софтуерната система.

5.2.3. Обектно-ориентиран дизайн

Ще бъдат прилагани принципите на обектно-ориентирания дизайн. Основните принципи, на който се обобщават в абривиатурата SOLID (Single responsibility, Open-closed, Liskov substitution, Interface segregation and Dependency inversion):


S	Принцип на конкретна отговорност (Single responsibility principle) Класовете трябва да имат конкретна, единична отговорност.
O	Принцип отворени/затворени (Open/closed principle) Софтуерните обекти трябва да бъдат отворени за разширяване и затворени за промяна.
L	Принцип на Лисков за заместването (Liskov substitution principle) Обектите в една програма трябва да бъдат заменени с инстанции от техни подтипове без да се променя коректността на програмата.
I	Принцип за разделяне на интерфейсите (Interface segregation principle) Много специфични за даден клиент интерфейси са по-добри от един интерфейс с общо предназначение.
D	Принцип за инверсия на зависимостите (Dependency inversion principle one) Нещата трябва да зависят от абстракциите, а не от конкретиката.

Тъй като дори и малките съвременни системи са изключително сложни за реализиране, има нужда от управление на обектно-ориентиран код, за да се избегне ненужната сложност. Тук се намесват шаблоните за дизайн, които са множество на добри практики за проектиране на кода.

В софтуерно инженерство, шаблонът за дизайн е решение за многократна употреба на проблеми, често срещащи се в даден контекст, в софтуерния дизайн. Дизайн-модел не е завършен дизайн, който може да се трансформира директно в код. Това е описание или шаблон за това как да се реши даден проблем, който може да се използва в много различни ситуации. Обектно-ориентираните шаблони за дизайн показват взаимоотношенията и взаимодействията между класовете или обектите, без да се уточняват окончателните класове или обекти, които ще бъдат въввлечени. Много шаблони предполагат обектно-ориентирано или по-общо променящо се състояние, поради което не са подходящи при езиците за функционално програмиране, при които обикновено данните се третират като непроменящи се.

Шаблоните за дизайн са модел, работещ с модули и връзки между тях. На по-високо ниво те са архитектурни модели, които са по-големи по обхват, обикновено описващи цялостната картина, следвана от системата.

Има различни видове шаблони за дизайн, които могат да бъдат използвани при разработката:



- 
- 1) Шаблоните за алгоритмични стратегии адресират въпросите, свързани със стратегиите от високо ниво, описващи как да се използват приложните характеристики на изчислителната платформа.
 - 2) Шаблоните за изчислителен дизайн адресират въпросите, свързани с ключовите изчислителни проблеми.
 - 3) Шаблоните за изпълнението адресират въпросите, свързани с изпълнението на приложението, включително стратегиите по изпълнение на система от задачи и работните блокове, осъществяващи синхронизацията между задачите.
 - 4) Шаблонните стратегии за имплементирането адресират проблемите, свързани с имплементирането на програмния код и организацията на програмата, както и общите структури от данни специфични за паралелното програмиране.
 - 5) Шаблоните за структурния дизайн адресират въпросите, свързани със структурите от високо ниво в разработваното приложение.

5.2.4. Моделиране с UML

В проекта ще се използва методика за моделиране, основаваща се на унифицирания език за моделиране (Unified Modeling Language – UML).

Unified Modeling Language (UML) е приложим към всички фази на реализирането на продукта. В частност, в сферата на системната архитектура, той допълва модела на проектиране с възможности за унифицирано моделиране и техники за право и обратно генериране (forward/reverse engineering). UML е приложим към техниките за софтуерна разработка, както и към техниките за бизнес анализ споменати по-горе. Той е стандартизиран език за моделиране с общо предназначение, разработен специално за сферата на обектно ориентираното разработване. Той включва в себе си множество графични правила и техники за създаване на визуални модели на обектно ориентирани софтуерни системи.

UML се използва за специфициране, визуализиране, конструиране и документиране на такива системи. Той предлага стандартизиран подход за визуализиране на проектите на разработваната система, включвайки елементи като:

- 
- действия;
 - актьори;
 - бизнес процеси;
 - схема на данните;
 - (логически) компоненти;
 - изрази на програмен език;
 - многократно използвани софтуерни компоненти.
- 

UML комбинира техники от моделирането на данните (entity relationship diagrams), бизнес моделирането (work flows), обектното проектиране и компонентното моделиране. Той може да бъде използван заедно с всички процеси от жизнения цикъл на проект за софтуерна разработка и е независим от различните технологии за реализация. UML синтезира в себе си методите на Booch, обектно ориентираните техники (OMT) и обектно ориентираното проектиране и разработка на софтуер (OOSE), сливайки ги в един общ и широко прилаган език за моделиране. UML се счита за стандарт в езиците за

моделиране, посредством който могат да бъдат моделирани конкурентни и разпределени системи. UML е де факто стандарт в индустрията и се развива под надзора на Object Management Group (OMG).

UML моделите могат да бъдат автоматично трансформирани в други представяния (например Java) посредством подобни на QVT езици за трансформиране. UML е разширяем с двата си механизма за настройване: профили и стереотипи.

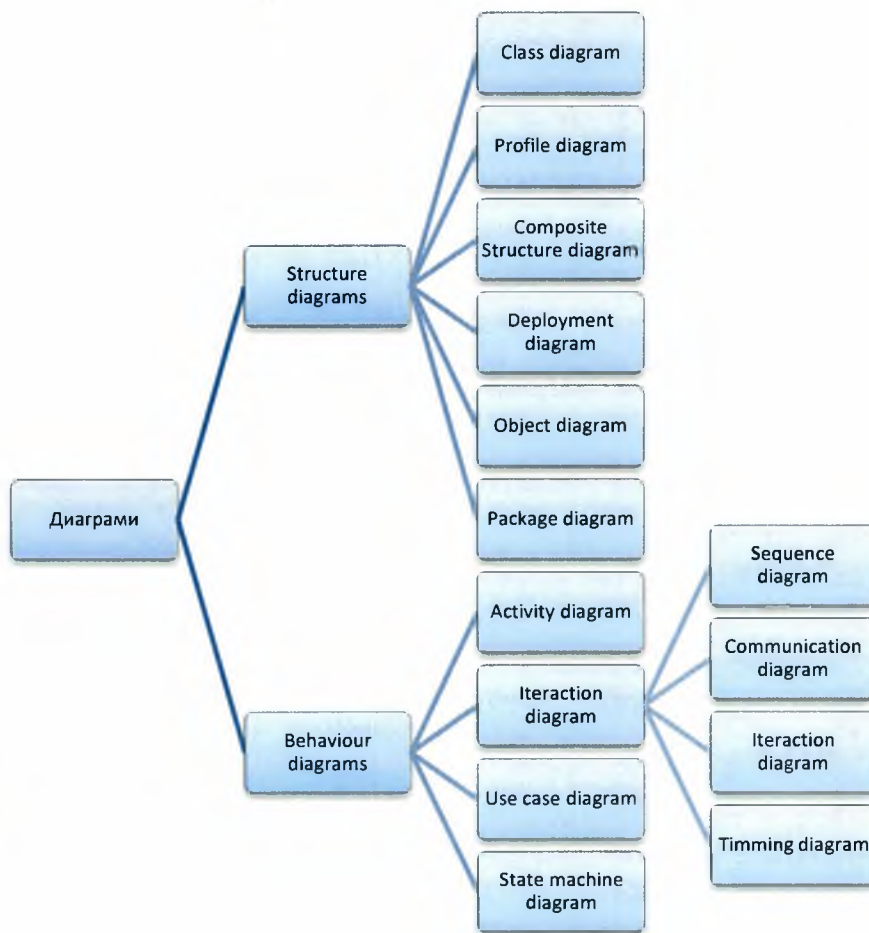
UML модели

Важно е да се направи разлика между UML модел и множество диаграми на дадена система. Диаграмата е частично графично представяне на модела на системата. Моделът съдържа информация, свързана с елементите и диаграмите му (като например текстови описания на изискванията на потребителите).

UML диаграмите представят два различни изгледа на модела на системата:

- 1) Статичен (или структурен) изглед – фокусът е върху статичната структура, като се използват обекти, атрибути, операции и взаимовръзки. Структурния изглед включва диаграма на класовете и диаграма на композитната структура.
- 2) Динамичен (или поведенчески) изглед – набляга на динамичното поведение на системата, показвайки взаимодействията между обектите и промените на вътрешните състояния на обектите. Този изглед включва диаграми като диаграма на последователностите, на действията и на състоянията.

UML 2.2 има 14 типа диаграми, разделени в двете категории. Седем от тях представят структурната информация, а останалите седем представят общите типове поведение, като четири от тях представят различни видове взаимодействие. Тези диаграми могат да бъдат категоризирани йерархично, по начина показан на следната клас диаграма:



Диаграми в UML 2.2

UML не ограничава UML типовете елементи да принадлежат към конкретна диаграма. Като правило всеки UML тип на елемент може да се появи на почти всички типове диаграми. Тази гъвкавост е частично ограничена в UML 2.0. UML профилите могат да дефинират допълнителни типове диаграми или да разширяват съществуващи диаграми с допълнителни елементи. За съвместимост с традиционните инженерни чертежи, UML диаграмите позволяват поставянето на бележки или описание на предназначението, ограничения или намерения.

5.2.5. Софтуер за моделиране на UML – Sparx Systems Enterprise Architect

Използваният продукт за документиране и моделиране ще бъде Sparx Systems Enterprise Architect - водещ продукт за моделиране с UML, поддържащ всички подходящи UML модели.

Със Sparx Enterprise Architect може да се създават и модели за стратегическо моделиране. Продуктът също поддържа моделиране на работните процеси с BPMN, както и организация и управления на изискванията, възможности за прототипиране на потребителски интерфейси и бяла дъска.

Продуктът може да бъде пригоден и за използване на модели, като ключовият за настоящия проект модел на бизнес дейностите.

Продуктът може да генерира документация на базата на поддържаните в него модели. Между голяма част от моделите може да се поддържат директни връзки.

5.2.6. Адаптиране на подхода за проектиране на софтуер
Дизайнът ще бъде съобразен с цялостната архитектура и специфики на>>>>.

За целите на пилотното тестване ще се използва част от цялостния модел на данните.

5.3. Подход за софтуерна разработка

Методологии и добри практики:	Итеративна разработка Унифициран процес за разработка Обектно-ориентирано програмиране Непрекъснатата интеграция
Средства и техники:	Scrum
Инструменти:	JetBrains IntelliJ Idea Sparx Systems Enterprise Architect Subversion Oracle Hudson Atlasian Jira
Източници:	www.ibm.com www.omg.org www.scrum.org
Обосновка:	Поради спецификите на проекта и кратките срокове за неговата реализация, проектът ще се изпълнява при ефективно съчетаване на класическия последователен подход (Waterfall) и итеративния (Agile) подход за реализация. Етапите на разработка ще бъдат групирани в итерации по фази. Предимство на този подход е предоставянето на използваем продукт в края на всяка итерация и навременното получаване на обратна връзка. Това дава възможност за гъвкавост при разработката при гарантиране на крайния резултат.

5.3.1. Процес за софтуерната разработка

Процесът на софтуерна разработка ще бъде цикличен и интерактивен посредством създаването на прототипи и одобрението им от потребителите. Това ще даде възможност за актуални прогнози и очаквания, като графика за предстоящата разработка ще бъде третиран като „жив“, променящ се документ, който отразява промените на изискванията и възникналите проблеми, съответно към намираните решения и успешното осигуряване на качеството.

Поради спецификите на проекта и кратките срокове за неговата реализация, проектът ще се изпълнява при ефективно съчетаване на класическия последователен подход (Waterfall) и итеративния (Agile) подход за реализация.



Когато програмната система се разработва от самото начало, процесът е същият като създаване на система по изисквания. Но след като веднъж системата придобие форма – или след като системата премине през първоначалния цикъл на разработка – всяка бъдеща доработка е процес на приспособяване на системата към новите или променени изисквания. Това се прилага през целия жизнен цикъл на системата.

Съществуват четири насочени към потребителя дейности по проектирането, за които е важно да започнат в най-ранната фаза на проекта:

- Разбиране и специфициране контекста на използване;
- Специфициране на потребителските и организационните изисквания;
- Даване на проектантски решения;
- Оценяване на дизайна в съпоставка с изискванията.

Итеративна разработка

Проект, в който се използва итеративна разработка, притежава жизнен цикъл състоящ се от няколко итерации. Всяка итерация включва множество не-строго последователни дейности в областта на бизнес моделирането, определянето на изискванията, анализ и проектиране, реализация, тестване и внедряване. Те са в различно съотношение в зависимост от това къде в жизнения цикъл на разработката е разположена итерацията.

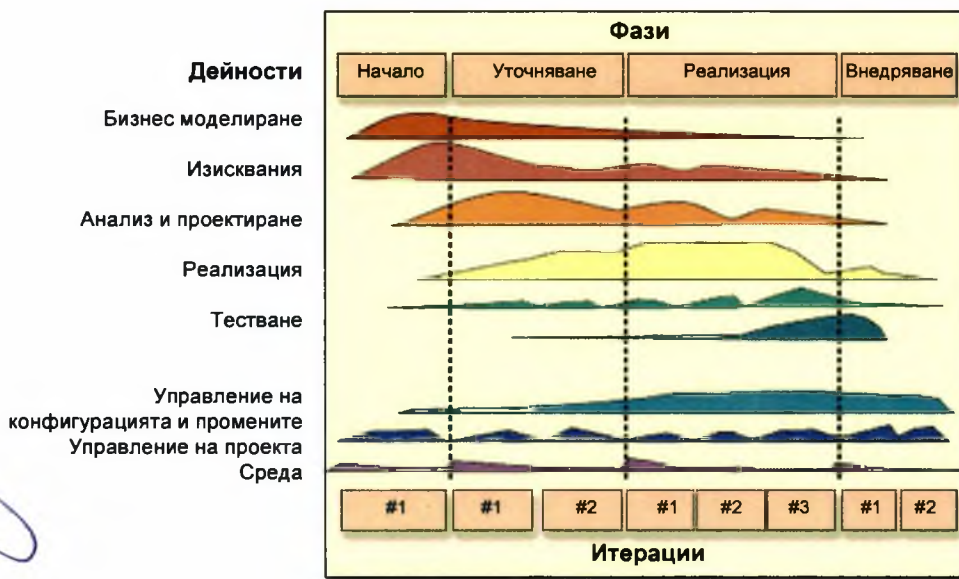
Итерациите в началната фаза и фазата на уточняване се фокусират върху управление на изискванията и дейности по проектирането. Итерациите във фазата на реализация се фокусират върху проектирането, разработката и тестването. Итерациите във фазата на внедряване се фокусират върху тестване и разпространение на системата. За всички итерации трябва да бъде прието, че са с определена времева рамка и графикът за изпълнението им да бъде фиксиран. Обхватът и съдържанието на итерациите трябва да се управлява активно така, че да се изпълни предвиденият график.

Итеративният подход превъзхожда линейния и каскадния (waterfall) поради няколко причини:

- Рисковете се намаляват рано т.к. елементите се интегрират прогресивно;
- Възприети са променящите се изисквания;

- Заложено е подобряването при изглаждането на продукта, което води до много по-качествен продукт;
- Организацията има възможността да се учат от този подход и да подобряват техните процеси;
- Повишена е възможността за повторно използване на компоненти;

Унифициран процес на разработка





Препоръчва се да бъде приложен Унифициран процес за разработка. Той дава възможност за организиран подход на възлагане на задачи и отговорности в рамките на организацията на разработчика. Неговата цел е да осигури разработка на висококачествен софтуер, който удовлетворява нуждите на своите потребители и е с предвидим график и бюджет.

Основните фази са:

Планиране (Подготовка) - Най-важната цел на началната фаза е да се постигне еднакво мислене на всички заинтересовани лица за целите през целия жизнен цикъл на проекта. Началната фаза е от решаващо значение най-вече за проекти, при които се започва разработка на нов продукт. При този вид проекти значителни рискове свързани с предметната област и изискванията могат да бъдат определени и предотвратени. За проекти, целящи разширяване на съществуващи системи, началната фаза е по-кратка, но също е фокусирана върху осигуряването на проекта, че ще се развива добре и че е възможен за изпълнение.

Проектиране - Целта на фазата е да начертае архитектурата на системата и да предостави основа за подробно проектиране и разработка във фазата на реализация. Системната архитектура се развива въз основа на най-значимите изисквания и на оценката на риска. Стабилността на архитектурата се оценява чрез един или няколко архитектурни прототипа.

Изграждане - Целта на фазата за реализация е да се изяснят неясните изисквания и да се завърши разработката на системата като се съблюдава начертаната архитектура. Фазата на реализация е един вид производствен процес, в който ударението е поставено върху управлението на ресурсите и контролирането на операциите така, че да се оптимизират разходите, графика и качеството.




Внедряване (Предаване) - Целта на фазата за Внедряване е да направи софтуера достъпен за неговите потребители. Фазата на внедряване може да премине през няколко итерации и включва тестване на продукта, подготовка за нови версии, корекции базирани на обратната връзка от потребителите. На този етап от жизнения цикъл, обратната връзка от потребителите трябва да е фокусирана основно върху прецизни настройки, конфигуриране, инсталиране и използваемост. Всички по-съществени структурни въпроси трябва да бъдат решени в по-ранните фази на проекта.

При изпълнението на проекта ще се използва методика за разработка на софтуер Rational Unified Process (RUP).

Управлението и изпълнението е организирано във фази, които могат да бъдат разделени на две или повече итерации. Основната цел на итерациите е да се постигне поетапно (итеративно) извършване на реализацията, тестването и внедряването. Итерациите по отделните фази обхващат една или повече дисциплини от цикъла за разработка на софтуерния продукт (бизнес моделиране, специфициране на изисквания, анализ и дизайн, реализация, управление на проекта, изграждане на среда за изпълнение).

Подход за приложение на RUP®

Методиката за разработка на софтуер, която ще приложим ще бъде IBM Rational Unified Process® (RUP®). Възникнали в хода на изпълнението въпроси, незасегнати в Техническото задание ще бъдат решавани съгласно изискванията на RUP®. За всички останали непокрити от Техническото задание или RUP® аспекти на проектното управление, като допълващи методи за управление, ще се придържаме към препоръките на ANSI/PMI 99-001-2008. Това са например Управление на доставките, управление на човешките ресурси и управление на финансовите разходи. В такива ситуации за управление на изискванията ще се придържаме към Business Analysis Body of Knowledge (BABOK®) Guide на IIBA®. По отношение на останалите дисциплини, аспектите непокрити от Техническото задание или RUP® ще бъдат изпълнявани под общоприети стандарти и методологии, като BPMN, UML, JEE, XML и други технологични стандарти.



Подходът ни за прилагане на RUP® ще бъде базиран на еволюционен цикъл за надграждане на съществуващо голямо приложение. Основни характеристики на прилагания от нас подход ще бъдат Итеративната разработка, Управление на изискванията, Архитектура на отделните компоненти, Визуално моделиране, Непрекъснатата проверка на качеството и Управление на промяната.

Начин на организация на итерациите

Съгласно RUP® жизнения цикъл на проекта ще бъде организиран във фази, които ще бъдат разделени на итерации. Дейностите ще бъдат разделени на фази по следния начин - Планиране (Inception), Проектиране (Elaboration), Изграждане (Construction) и Внедряване (Transition).

Целите, които си поставяме в края на Планирането е детайлизиран план за изпълнение на жизнения цикъл. В края на Проектирането, целта е да бъде изготвена архитектура и дизайн на системата. В края на Изграждането трябва да бъде постигнат базов оперативен капацитет на системите. В края на Внедряването трябва да бъде въведена в експлоатация окончателната версия на продукта.

В рамките на всяка една от фазите са обособени една или повече итерации. Изпълняваните дейности са свързани с определени отчетни продукти. Съгласуван списък с отчетни продукти и описание на съдържанието им в съответствие с RUP ще бъдат заложили в Плана за разработка на софтуерния продукт, който ще бъде утвърден от Възложителя в края на фаза Планиране. Допълнителна информация за конкретните фази на изпълнение:

- Планиране/Inception – В рамките на тази фаза ще си извършват предимно дейности по планиране от всички работни процеси. Паралелно с тях ще стартират дейности свързани с изпълнение на задачи по Проектиране, Изграждане и дори Внедряване с инициране на планиране на внедряването. Фазата на планиране ще завърши с предаването на изискваните съгласно Техническото задание отчетни продукти - План на проекта, План за управление на качеството, Списък на рисковете, Спецификация на функционалните изисквания, Спецификация на допълнителните изисквания и План за следващата итерация. Дейностите по планиране ще продължат, като преди началото на всяка итерация ще се извършва оценка на постигнатото в предходната итерация и планиране за следващата итерация. В началото на всяка фаза ще се извършва планиране на итерациите по фазите. Дейността по планиране ще бъде осъществена по отворен за проектния екип начин, така че да служат ефективно като средство за управление на риска и промяната.
- Проектиране/Elaboration – Ще бъдат разработени прототипи за доказване на изпълнимостта на продукта. Проектирането е предмет на планиране и в този смисъл то винаги следва дейностите по планиране. Същинското проектиране ще започне с приключването на фаза Планиране. Проектирането ще бъде извършено в началото на проекта в широчина на обхвата, като в рамките на всяка итерация ще бъде конкретизирано по компоненти. По време на проектирането ще се изпълняват и дейности от Изграждане и Внедряване. Проектирането изисква високо ниво на комуникация и съвместна работа с експерти на Възложителя. То ще завърши с предаване на отчетните материали за фазата, съгласно Техническото задание.
- Изграждане/Construction – Основните цели на тази фаза са разработката на софтуера и създаването на използвани версии (алфа, бета и други тестови версии) в предвидените срокове и с необходимото качество. През тази фаза ще се приключи с дизайна, реализацията и алфа тестовите на цялата заявена функционалност с цел получаване на завършен продукт, годен за предаване на бъдещите му потребители. Извършва се тестване на база дефинираните критерии за оценка. То ще започне с приемането на Функционално-техническата спецификация. Изграждането е изключително итеративен процес. То ще се изпълнява по компоненти (капсуловани кратки функционалности), като няколко компоненти ще се изпълняват паралелно. Точният график на итерациите ще бъде специфициран в Плана за разработка на софтуерния продукт. Всяка итерация ще бъде предхождана от дейности по Планиране и дейности по Проектиране. Итерациите ще завършват с дейности по оценка на изпълнението и с дейности по вътрешно тестване. Функционално завършени компоненти ще бъдат предавани на Възложителя за приемно тестване (част от фазата Внедряване). В този смисъл Изграждането ще завърши с получаване на одобрение от Възложителя за въвеждане на компонента в експлоатация, т.е. Внедряване.
- Внедряване/Transition – Основните цели на фаза Внедряване са: Провеждане на бета тестове за валидиране на системата от гледна точка на потребителските изисквания; Мигриране на стари данни, ако е необходимо; Специфични за етапа на внедряване дейности по инсталиране; Фини настройки като отстраняване на дефекти, подобряване на производителността и използваемостта на системата; Оценка на индикаторите за внедряване на база критериите за приемане на продукта; Получаване съгласие от Възложителя, че индикаторите за внедряване са изпълнени и съответстват на критериите за оценка. Внедряването на даден компонент ще започне след получаване на одобрение от Възложителя за въвеждане на компонента в експлоатация. Внедряването ще завърши с

въвеждането в експлоатация на продуктите и получаването на съответното потвърждение от Възложителя.

Изборът ни на традиционен подход за разработка, базиран на Rational Unified Process се обосновава на следните факти:

- Поръчката се възлага чрез договор с фиксирани обхват, цена и срок. Предложеният подход съвпада с изискванията за етапи на изпълнение, съгласно Техническото задание.
- Срокът за изпълнение е изключително кратък, което предполага интензивен анализ на изискванията и проектиране в началото на изпълнението, както и придържане към твърд график по време на цялото изпълнение.
- Световно-утвърден и доказан подход за разработка.
- Позволява итеративно предоставяне на компоненти от разработката.

5.3.2. Характеристики на процеса за софтуерна разработка

На база на изготвени анализи, моделирани процеси и спецификации на програмни интерфейси за всеки елемент на системата, Изпълнителят ще започне неговата разработка, като изпълнява най-малко поддейностите, които са приложими, както са дефинирани в следващите секции на ТС.

Разработка:

- Разработка на схеми за данни / структура на базата/ите данни;
- Разработка на изходен код за имплементиране на бизнес-логика
- Разработка / генериране на изходен код за автоматизирани процедурни тестове (Unit Tests)
- Разработка / генериране на изходен код за програмни интерфейси и клиентска интеграция;
- Идентифициране и специфициране на нужди от промени в програмни интерфейси (APIs) на други елементи или компоненти, които са необходими за реализация на разработваните функционалности и процеси, които зависят от тях;
- Разработка на екранни форми за електронно представяне на процесите и услугите (където е приложимо);

Тестване:

- Разработка на тестови сценарии за проверка на реализацията на системата;
- Тестване на разработената функционалност по съответния компонент;
- Документиране на резултатите от проведените тестове.

Представяне:

- Представяне на Възложителя на разработени функционалности чрез изпълнение на предварително дефинирани тестови сценарии;

Изпълнението на дейностите по разработка ще бъдат планирани, както по елементи, така и по дефинирани периодични итерации, включващи разработка и тестване на отделните елементи и компоненти. Трябва да бъде предвиден и координационен механизъм и процеси за обмен на спецификации за промени в програмни интерфейси между отделните екипи, изпълняващи дейности по разработка на елементи от системата, ако е приложимо.

Тестване и представяне на резултатите на Възложителя ще се извършва за всяка итерация, в оперативен порядък. В случай, че Възложителят има изисквания за корекции по реализираната функционалност, те трябва да бъдат отразени от Изпълнителя до представяне на следващата итерация по разработка на съответния компонент.

[Handwritten signature]

В случай, че Възложителят няма забележки, итерацията се приема и се интегрира в софтуерното решение.

Промяна:

- Стъпка „Промяна“ се извършва само в случаите, в които Възложителят е имал забележки и корекции по реализираната функционалност в една итерация, описана в предходната стъпка.

Интегриране:

- Стъпка „Интегриране“ обхваща дейностите, свързани с включване на разработената и одобрена от Възложителя функционалност в един компонент.

Документиране:

- Стъпка „Документиране“ включва генериране на документация за текущата (build) версия на програмните интерфейси за интеграция на отделните елементи, съгласно изискванията за документиране на програмни интерфейси, описани в т. 7.9.2 от ТС.

Като резултат от дейност „Разработка“ Възложителят ще получи:

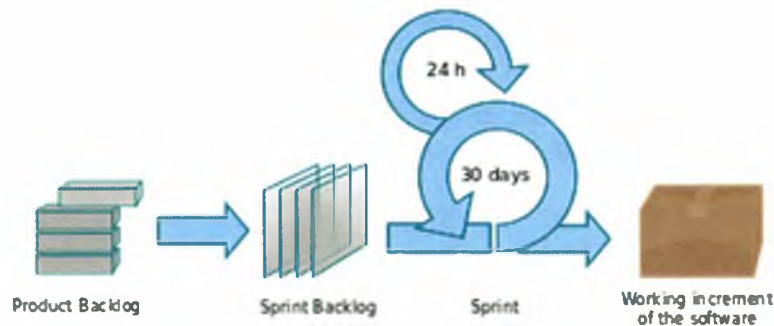
- Работеща функционалност на системата (система, подсистема, модул, компонент);
- Изходен код с версия за публикуване (Release Version);
- Тестови сценарии и доклади от проведени тестове.

[Handwritten signature]

5.3.3. Вътрешно-екипна координация в рамките на итерациите

Вътрешно екипната координация ще бъде реализирана чрез прилагането на Scrum. Scrum е итеративна, инкрементална рамка за управление на проекти.

[Handwritten signature]



Макар че подходът на Scrum е първоначално предложен за управление на проектите за разработване на продукти, той се фокусира с времето върху управлението на софтуерни проекти и може да бъде използван, за да задвижва екипи за софтуерна поддръжка или като общ подход за проектов / програмен мениджмънт. Второто определение го прави доста ценно и нужно за България и българското образование, т. к. в страната ни има недостатъчно на брой специалисти от световно ниво по “product ownership” и “product development”. Това е предпоставка за факта, че често разработваме софтуер, който някой друг продава, или за който някой друг вече е намерил клиентите.

Scrum процесът се състои от отделни спринтове. Спринтовете могат да имат продължителност от една седмица до четири седмици. В края на всеки спринт, екипът разполага с работеща версия на продукта, която включва всички готови задачи от backlog-а.

[Handwritten signature]

SCRUM позволява организирането на самоорганизиращи се екипи като стимулира това всички членове на екипа да се намират на едно и също място и да си комуникират на живо. Ключов принцип на SCRUM е това, че се приема още в самото начало на проекта, че изискванията няма как да са пълни и напълно разбрани. Т.е. очакват се нововъведения от клиента – промяна на желанията му. SCRUM се фокусира върху способността на екипа да доставя бързо и да е готов да отговори бързо на неочакваните промени. Това е положителна черта за SCRUM, защото резките промени не могат да бъдат добре оборени с традиционните отгатващ или планиращ методи. Както в другите гъвкави методологии за управление на проекти, SCRUM може да се имплементира чрез различни видове инструменти. Най-много се използват спредшийтове. В тях се правят така наречените SCRUM „артефакти“, като sprint backlog-а. Други организации имплементират SCRUM с бели дъски, лепящи се листчета и хартиени документи.

Спринтът е най-малката единица време за разработване. Спринтовете са с постоянна дължина от 1 седмица до 1 месец. Всеки спринт е опит за подобрене, вкаран във фиксирани времеви рамки. Преди всеки спринт има среща за планиране на спринта. На нея се поставят измеримите цели за спринта и се идентифицират задачите, които ще бъдат свършени в неговите рамки. По време на всеки спринт екипът създава завършени парчета от даден продукт. Дейностите за всеки спринт се описват и взимат от „продуктовата опора“ или на английски „product backlog“. Често тези дейности са описани като характеристики, които продукта трябва да има и да бъдат постигнати за спринта. Т.е. product backlog е приоритизиран списък на изискванията. Какво от списъка да влезе в даден спринт се решава на планиращата среща преди спринта. Продуктовият собственик уведомява екипа кои части от списъка с изисквания иска да бъдат свършени на предстоящия спринт. Развойният екип преглежда, обсъжда, решава и записва в Sprint Backlog кои от тези изисквания и цели ще успее да изпълни на предстоящия спринт. Sprint Backlog е собственост на развойния екип. Целите, вписани в този документ не трябва да бъдат променяни по време на спринта. За разработването се определя фиксирана продължителност, такава, че спринтът да свърши навреме. Изискванията, които не бъдат удовлетворени за спринта се изключват и връщат към product backlog. След като спринтът е изпълнен, екипът демонстрира как се използва софтуерът.


По време на спринта всеки ден се провеждат т.нар. правостоящи срещи (stand-up meetings). Тези срещи продължават от 5 до 15 минути и се провеждат всеки ден в определен час. На срещата всеки от екипа абсолютно неформално разказва за три неща: какво е работил предишния ден, какво планира за предстоящия ден и какви проблеми е срещнал, които му пречат да работи.

Характерно за ежедневните срещи е следното:


- Срещата започва точно навреме.
- Мястото и часът са едни и същи всеки ден.
- Срещата трябва да се вмести точно в 15 минути.
- Може всички, но главно говорят основните роли в екипа.
- На срещата се обновява и backlog-ът като се отбелязва свършената работа. Ако се идентифицират някакви проблеми, те се решават колективно. Важно е да се отбележи, че това не са срещи за отчет пред ръководството, а за синхронизация (самоорганизация) на екипа и разкриване на потенциални пречки в работата.

Специфични практики:

- Ежедневни правостоящи срещи (Stand-up meetings)
- Backlog: списък със задачите за текущия спринт и тяхното състояние

- 
- Самоорганизиращ се екип: екипът не следва предварително раздадени задачи, а всеки негов член се стреми да допринесе за постигне целите на спринта - всеки ден всеки си взема задачи, за които отговаря
 - Работни срещи и обсъждания с клиента и с екипа след всеки спринт


Срещи след спринта:

- Среща за обзор на спринта: Преглежда се и работата, която е била свършена, и тази която не е. Представя се свършената работа на клиентите (показва се демо). - Несвършената работа няма как да бъде представена, но се описва. - Срещата трябва да се вмести в 4 часа.
 - Спринт Ретроспекция - Всички членове на екипа си припомнят и обсъждат миналия спринт.
- 

5.3.4. Обектно-ориентирано програмиране

Обектно-ориентираното програмиране (ООР) е парадигмата за програмиране, използваща "обекти" - структури от данни, състояща се от полета с данни и методи, в едно с техните взаимодействия. Използва се за проектиране на приложения и компютърни програми. Техниките за програмиране могат да включват функции като абстракция на данни, капсулиране, съобщения, модулност, полиморфизъм и наследяване. Много съвременни езици за програмиране, които сега поддържат ООР, поне като опция.

Дори простите ООР програми могат да представляват един "дълъг" списък от команди. По-сложните програми групират по-малки участъци от код във функции или така наречените подпрограми, всяка от които може да изпълнява определена задача. С дизайн от този вид става обичайна практика за някои от данни на програмата да са „глобални“, т.е. достъпни от всяка част на програмата. Когато програмите станат по-големи, и е позволено всяка функция да променя всяка част от данните, се оказва, че грешките могат да имат сериозни последици.



За разлика от този подход, обектно-ориентираното програмиране насърчава програмистите да поставят данните, където им е мястото, т.е. да не са директно достъпни от останалата част на програмата. Вместо това, данните са достъпни чрез използване на написани за целта функции, често наричани методи, които или вървят в пакет с данните, или се наследяват от „класа на обекта“. Те действат като посредници за извличане или промяна на данните, които контролират. Програмирането е конструкция, която комбинира данни и методи за достъп и управление на тези данни, нарича се обект. Практиката да се използват подпрограми за достъп и промяна на данни е използвана и в необектно ориентираното програмиране.

Обектно-ориентираните програми обикновено съдържат различни видове обекти, всеки вид, съответстващ на определен вид комплексни данни, които ще се управляват или отговарят на обект или понятие от реалния свят, като например банкова сметка, хоккей играч или булдозер. Една програма може да съдържа няколко копия на всеки тип обект, по един за всеки обект от реалния свят. Например, може да има един обект на банкова сметка за всяка истинска сметка в определена банка. Всяко копие на обекта на банкова сметка, ще бъде с еднакви методи за манипулиране или четене на данни, но данните вътре във всеки обект ще се различават и ще отразяват различната историята на всяка сметка.

Обектът може да се разглежда като опаковка на данните, заедно с набор от функции, предназначени да гарантират, че данните се използват по подходящ начин и да съдейства в тази употреба. Методи на обекта, обикновено включват проверки и предпазни мерки, които са специфични за данните, които се съдържат в обекта. Един обект може да предлагат прости за използване, стандартизирани методи за извършване на определени операции върху данните си, като същевременно прикрива спецификата на това как се изпълняват тези задачи. По този начин могат да бъдат правени промени във вътрешната структура или методите на обект, без да се изисква модифициране на останалата част от програмата.

Handwritten signature

Този подход може също така да бъде използван и за предлагане на стандартизирани методи в рамките на различните видове обекти. Като пример, няколко различни видове обекти може да предложи методи за печат. Всеки тип на обекта може да реализира този метод за печат по различен начин, отразяващ различни видове данни, но същевременно различните методи за печат могат да са кръстени по еднакъв и стандартизиран начин на други места в програмата. Тези характеристики стават особено полезни, когато повече от един програмист разработва код в проекта или когато целта е за повторно използване на код между проектите.

Ориентацията към обекти е свързана с поддържането на качеството на софтуера. Тя е насочена към общи проблеми, като набляга на създаването на дискретни, подходящи за многократна употреба модули от програмна логика. Технологията се фокусира върху данните, вместо върху процесите, като програмата се състои от самостоятелни модули („класове“), всяко копие, на които („обекти“) съдържа цялата информация, необходима, за да може да манипулира своята собствена структура от данни („членове“). Това е в контраст на съществуващите техники за модулно програмиране, които са били доминиращи в продължение на много години, които се фокусират върху функциите на един модул, а не върху конкретните данни, но по подобен начин предлагат възможност за повторно използване на код, самостоятелни са и са подходящи за многократна употреба, дават възможност на сътрудничество чрез използването на свързани модули (подпрограми). Това е по-традиционния подход, който все още продължава, да се използва. При него данните и поведението им се разглеждат отделно едни от други.

Handwritten mark

Обектно-ориентираната програма може да се разглежда като съвкупност от взаимодействащи си обекти, за разлика от конвенционалния модел, в които програмата се разглежда като списък от задачи (подпрограми) да изпълнява. В обектно-ориентирания подход, всеки обект е в състояние да получава съобщения, да обработва данни и да изпраща съобщения към други обекти. Всеки обект може да се разглежда като независима „машина“ с ясно изразена роля или отговорност. Действията (или „методите“) на тези обекти са тясно свързани с обекта. За пример, ООП структури от данни са склонни да „носят своите оператори заедно с тях“ (или най-малкото да ги „наследяват“ от подобни обекти или класове) - с изключение на случаите, когато те трябва да бъдат сериализирани.

За да бъде един програмен език обектно-ориентиран, той трябва не само да позволява работа с класове и обекти, но и трябва да дава възможност за имплементирането и използването на принципите и концепциите на ООП:

Handwritten mark

- Наследяване (Inheritance) - как йерархиите от класове подобряват четимостта на кода и позволяват преизползване на функционалност;
- Абстракция (Abstraction) – виждането на даден обект само от гледната точка, която ни интересува и да игнорираме всички останали детайли;
- Капсулация (Encapsulation) – скриване на ненужните детайли в нашите класове и предоставяне на прост и ясен интерфейс за работа с тях;

Полиморфизъм (Polymorphism) - работа по еднакъв начин с различни обекти, които дефинират специфична имплементация на някакво абстрактно поведение.

5.3.5. Дейности за интегриране на функционалности/компоненти в системата

В етапа на разработка, стъпка „Интегриране“ обхваща дейностите, свързани с включване на разработената и одобрена от Възложителя функционалност в един компонент.

Handwritten signature

Интеграцията е процес, при който се комбинират отделни софтуерни компоненти в единна система.

Интеграцията е и форма на тест – Тъй като тя се извършва след като разработката и тестването на отделните компоненти е приключило. Също така тя е свързана и с интеграционните тестове. Тя е достатъчно сложна, за да се разглежда като независима част от проекта – Правилната и внимателна интеграция води след себе си :

- По-лесна диагностика;
- По-малко дефекти;
- Съкратено време за завършването на продукта;
- Подобро качество на кода;
- Повишени шансове проекта да приключи успешно;
- По-малко документация;

и много други.

В тази дейност се очаква да бъдат:

- Изградени съответните интеграционни интерфейси;
- Интегрирани съответните компоненти;
- Извършени тестове за интеграция.

Като резултат от дейност Интеграция Възложителят ще получи:

- Успешно интегрирани компоненти на системата;
- Доклади от проведени тестови изпитания и резултатите от тях.

Целият хардуер, компоненти, модули, части и софтуерни лицензи на специфицираното и предложено от Изпълнителя оборудване, ще бъдат сглобени, тествани, инсталирани и ще се валидира тяхната функционалност и работоспособност в работно помещение на Възложителя.

Съществуват два основни подхода за интеграция на компонентите:

- Единна интеграция;
- Нарастваща (инкрементална) интеграция.

Единна интеграция

Единната интеграция включва следните стъпки:

- 1) Дизайн, конструиране на кода и тестове
- 2) Всичко направено се събира заедно
- 3) Извършване на тестове върху така получената пълна система

Този вид интеграция дава предимство, че предварително могат да се планират и обхванат всички интеграционни задачи, но има и редица недостатъци, които го правят приложим само за по-малки системи.

- Не е ясно от къде идват проблемите;
- Този тип интеграция не може да започне в ранна фаза на проекта. Трябва да се изчака почти до самия му край.

Тъй като съществува голям брой класове, които до сега не са работили заедно, когато възникне грешка няма как да я идентифицираме лесно. Не е ясно, дали даден клас не е тестван добре, дали е грешка

заради интерфейса, по който комуникират два компонента или пък е грешка от взаимодействието на два класа. Този недостатък до някъде се компенсира от факта, че всички проблеми се появяват едновременно – по този начин програмистите трябва да се справят не само с проблеми от взаимодействието на компонентите, но и с проблеми, които се появяват от поради наличието на няколко други проблема.

Нарастваща интеграция

При нея процесът протича по следния начин:

- 1) Разработва се малка част от системата – В общия случай за всеки проект се подхожда различно. Началния код, който се разработва може да е най-лесната или най-трудната част, както и някаква комбинация между тях.
- 2) Разработва се нов клас, тества се и т.н.
- 3) Новият клас се добавя към вече съществуващата част и така получената комбинация се тества отново – Ако всичко е наред с тестовете и е останало още за доразработване, се преминава обратно в стъпка 2.

Нарастващата интеграция дава следните предимства:

- Не е нужно добавянето да става клас по клас;
- Грешките се откриват по-лесно - Или интерфейсът за взаимодействие предизвиква проблема или е взаимодействието на новия клас, с някой от вече интегрираните класове. И двата случая е ясно къде да трябва да се погледне, за да се намери проблема.
- Системата е видима в ранна фаза – Това подобрява морала на програмистите и атмосферата на работа, понеже виждат как системата се изгражда и работи. С единната интеграция винаги остава съмнението, че когато нещата се съберат заедно, нищо няма да сработи.
- Подобен контрол над прогреса – Много по-лесно е да определим как се движи проекта и дали е по график, когато виждаме 50% от него вече работещ.
- Клиентът е по-щастлив – Клиентите обичат да виждат признаци на прогрес. Нарастващата интеграция непрестанно им предоставя факти по напредъка.
- Модулите на системата се тестват по-пълноценно – Първо, тестовете почват в много по-ранна фаза. Второ, всеки клас се тества много по-често (с всяко добавяне на следващ модул).
- Системата се изгражда за по-кратко време – Ако интеграцията е планирана правилно, работата може да започне по някаква част от проекта, докато друга част все още е на етап дизайн. Това няма да намали общото време за реализация, но позволява част от работата да се свърши в паралел.

Съществуват различни видове стратегии за прилагане на нарастваща интеграция:

- **Top-Down интеграция** - При Top-Down интеграцията, класът стоящ най-високо в йерархията се пише и интегрира пръв. Започва се с базовия клас на системата (съдържащ main(), WinMain() и т.н.) Останалите класове често се пишат само като шаблони (съдържащи само декларациите на методите) и в последствие се дописват и добавят. Тъй като чистия Top-Down подход рядко се използва, негов по-практичен вариант е вертикалната Top-Down интеграция. Тя елиминира повечето проблеми на предишния подход и е често използвана в практиката.
- **Bottom-Up интеграция** - При Bottom-Up интеграцията, добавянето на класове върви в обратен ред (от долу нагоре). Основният проблем при Bottom-Up интеграцията е, че системните интерфейси в най-високото ниво се интегрират последни. Ако в системата има концептуална грешка или проблем свързан с дизайна, той остава скрит почти до края на интеграцията. Освен това, ако в такъв момент дизайнът се смени, може да се наложи част от класовете в ниските нива

[Handwritten signature]

на йерархията да се пренапишат. Аналогично, за избягване на недостатъците на Bottom-Up, се използва вертикална Bottom-Up интеграция, точно както при Top-Down.

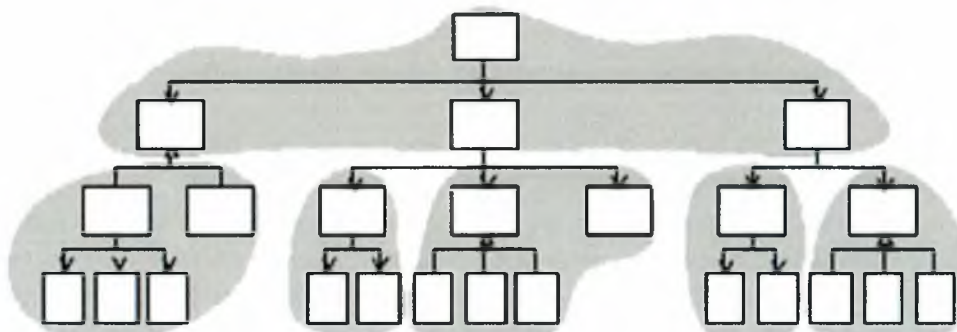
- **Sandwich интеграция** - Заради проблемите, които могат да възникнат при Top-Down и Bottom-Up интеграцията, някои експерти препоръчват интеграционния процес да протече така: първо се интегрират класовете от най-високо ниво, после тези от най-ниско ниво, а свързващото ги звено остава за последната фаза. Това е реалистичен и много практичен метод за интеграция.
- **Рисково-ориентирана интеграция** - Основният принцип при рисково ориентираната интеграция е "най-трудната част се интегрира първа". В общия случай, при този тип интеграция се интегрират класовете стоящи най-високо и най-ниско в йерархията. Този подход много прилича на Sandwich интеграцията, тъй като изследванията показват, че най-рисковите нива са в най-горната и най-долна част на йерархията.
- **Функционална интеграция** - При функционалната интеграция основната идея е, че основният интеграционен елемент е функцията (в смисъл на възможност или характеристика на приложението). Например, ако пишем текстообработващо приложение една функция може да бъде Fonts диалога, класът за изобразяване на ефекти върху шрифтовете, менютата и т.н. Ако някоя от функциите, които ще интегрираме се състои от множество класове, към тях отново може да се приложи стратегия за нарастваща интеграция. Т.е. стратегиите за нарастваща интеграция могат да се използват рекурсивно.
- **Т-образна интеграция** - При Т-образната интеграция се изгражда и интегрира "дълбок срез" вертикално през целия проект. По този начин се проверява дали предположенията направени по време на проектирането на архитектурата са коректни. След като тази част приключи, всички останали модули се интегрират хоризонтално.

[Handwritten scribble]

Няма една "най-добра" методология за интеграция. Някоя от тях не е универсална и не трябва да се спазва стриктно. Винаги се търси метод, по-скоро базиран на някоя от стратегиите, така че той да пасне най-добре на самия проект.

5.3.6. Стратегия за интегриране на функционалности/компоненти в системата

В етапа на анализ и проектиране ще бъде разработена детайлна системна и софтуерна архитектура. Като част от архитектурата ще бъдат поставени стандарти за интеграция на компонентите, както и ще бъдат дефинирани основните интеграционни интерфейси.





Функция 2 Функция 3 Функция 4 Функция 5 Функция 6

[Handwritten signature]

При разработката препоръчваме прилагането на стратегия за нарастваща функционална интеграция, която дава следните предимства:


- Редукция на поддържащия код – Добавянето на новата функция към проекта не е свързано с модификация или създаване на поддържащ код. Подлежащата на интеграция функцията е независима и съдържа всичко нужно за работата си.

[Handwritten signature]

- 
- 
- Новоинтегрирания код води до нова функционалност – Това показва, че проекта върви напред. От една страна повдига духа на разработчиците, а освен това подобрява и отношенията с клиента.
 - Добра съвместимост с ООП – Тъй като най-често обектите са свързани с някаква възможност или функционалност, този подход действа много естествено в обектно-ориентираните системи.

5.3.7. Дневни билдове и тестове

Добра практика при прилагането на методологията за интеграция е извършването на дневни билдове в комбинация с тестване, които водят до следните ползи:

- 
- Намален риск на качеството – Отстранява се възможността някоя грешна интеграция да предизвика сериозни проблеми в системата. Проекта веднъж докаран до добро, работещо състояние се държи на това ниво и всеки ден се проверява, че всичко е наред.
 - По-лесна диагностика на дефектите – Ако проекта се проверява всеки ден и един от дните спре, веднага е ясно, че проблемът е в код добавен в последния ден.
 - Подобен морал на разработчиците – Самият факт, че продуктът работи много силно влияе върху морала на разработчиците. Чрез дневните билдове те се уверяват всеки ден, че техния продукт работи.
 - Изважда на преден план “невидими” задачи – Страничен ефект от ежедневната интеграция е факта, че тя изважда на преден план доста работа, която иначе може да се пропусне и да се появи чак в края на проекта. Екипи, които не са ползвали дневни билдове, първия път винаги се оплакват, че дневните билдове ги бавят. Това не е така, просто им се налага да отделят ежедневно ресурси за интеграция на кода, които иначе биха им се струпали накрая.

При изпълнението на дневни билдове и тестове ще се прилагат следните принципи:

- Разработчиците трябва сами да тестват кода си, преди да го добавят към проекта – Един добър метод е всеки разработчик да си прави личен билд, да го тества и чак тогава да добави промените си към целия проект.
- Централно хранилище за код – Основна част от успехите на дневните билдове идват от факта, че знаем кои билдове са добри и кои са пропаднали. При малки и средни проекти за да се пази история се използва система за контрол на версиите. При големите проекти обикновено само билд групата има права да добавя код в главната системата за контрол на версиите.
- „Наказания“ за “счупване” на билда – На разработчиците трябва да е ясно, че дневния билд винаги трябва да работи. Пропадането му е голямо изключение (никога практика) и всеки трябва да счита с най-висок приоритет задача по възстановяване на пропаднал билд. В Microsoft, разработчиците от Windows 2000 и Office групата носят денонощно пейджъри и биват викани и в 3:00 през нощта, ако се окаже че техният код проваля билда.
- Билдовете приключват сутрин – Добра идея е билдовете да минават нощем, тестовете да се правят рано сутринта и процеса да приключва още преди обяд. Така тестерите могат всяка сутрин да тестват с нов билд. Сутрин колегите се откриват най-лесно (в случаи, когато с билда има проблем). Викането на хора посред нощ поставя излишно напрежение в екипа.
- Дневни билдове и тестове трябва да се извършват без значение натовареността – Когато времето напредва и напрежението нараства, разработчиците губят дисциплина. Никога не трябва да се допуска това да влияе върху дневните билдове и тестове.

5.3.8. Управление на версиите

Непрекъснатата интеграция

В софтуерно инженерство, непрекъснатата интеграция (CI) осъществява непрекъснат процес на прилагане на контрол на качеството - малки усилия, които се прилагат често. Непрекъснатата интеграция има за цел да подобри качеството на софтуера и да намали времето, необходимо той да бъде доставен, чрез заместване на традиционната практика на прилагане на контрол на качеството, след завършване разработката.

При навлизането на промяна, разработчиците вземат копие на текущата кодова база, върху която да работят. Когато други разработчици направят промени в изходния код, те го изпращат в хранилището на версиите. Когато разработчиците представят код на хранилището, те трябва първо да актуализират своя код, за да отразяват промените в хранилището, тъй като там е най-актуалното копие. Колкото повече промени съдържа хранилището, толкова повече работа ще свършат разработчиците, преди да предадат своите собствени промени.

В крайна сметка, хранилището може да стане толкова различно от основната линия на разработчиците, че те да попаднат в така наречения "интеграционен ад", където времето, необходимо да се интегрират надхвърля времето, необходимо, за да направят промените. В най-лошия сценарий, разработчиците могат да изхвърлят своите промени и да извършат наново цялата работа.

Непрекъснатата интеграция предполага интегриране рано и често, така че да се избегнат капаните на "интеграционния ад". Практиката има за цел да намали преработките и по този начин да се намалят разходите и времето.

Непрекъснатата интеграция - практика на често интегриране на нов или променен код в хранилище на съществуващия код - ще се случва достатъчно често, така че да не остава прозорец между качване на код и компилиране, и поради това да не може да възникне грешка без програмиста да я забележи и коригира незабавно. Нормалната практика е да се задействат при всяко качване на код, а не периодично. Тези практики в среда от много програмисти с бързи комити (включване на програмен код в интеграционната среда) се преобразува в такава, при която се задава кратък интервал след всяко качване на код и чак след това се пуска компилирането.

За да постигне своите цели, непрекъснатата интеграция разчита на следните принципи:

- 1) Поддържане на хранилище на код - Тази практика се застъпва за използването на система за контрол на ревизии на изходния код на проекта. Всички артефакти, които са необходими за изграждане на проекта ще бъдат поставени в хранилището. Важна практика е да бъде възможно системата да се компилира от пряко извлечен код от системата без да е необходимо удовлетворяването на допълнителни зависимости. Използването на клонове ще бъде сведено до минимум. Промените ще бъдат бързо интегрирани вместо да се създават множество разклонения. Основната линия (trunc) ще бъде място за работеща версия на софтуера.
- 2) Автоматизиране на компилацията – ще може да се извърши с една команда. Автоматизация на компилацията ще включва автоматизиране на интеграция, която често включва внедряване в производствената среда. В много случаи, скрипта за компилация не само компилира изпълнимите файлове, но също така генерира документация, страниците на уебсайта, статистика и окончателната дистрибуция (например Windows MSI файлове, RPM или DEB файлове).
- 3) Компилацията ще може да се тества самостоятелно - След като кодът е компилиран, всички тестове би трябвало да се стартират, за да потвърдят, че той се държи така, както програмистите очакват да се държи.
- 4) Ежедневно качване в базовата линия - чрез редовно качване в базовата линия, всеки програмист може да се намали броят на конфликтните промени. Седмично качване



води до риск от увеличаване на конфликтните промени, които може да са трудни за разрешаване. Ранните, малки конфликти в дадена областта карат програмистите да общуват относно промените, които правят.

- 5) Всяко качване (в базовата линия) ще се компилира - Системата трябва компилира текущата версия, за да определи дали направените промени са правилно интегрирани. Честа практика е да се използват автоматизирани средства за непрекъсната интеграция, въпреки че това може да се направи и ръчно. За много хора, непрекъснатата интеграция е синоним на използването на сървър за автоматизирана непрекъсната интеграция, който автоматично стартира процеса по компилиране.
- 6) Компиляцията ще бъде поддържана бърза – ще се изпълнява бързо за да може, ако възникне проблем с интегрирането да бъде идентифициран бързо.
- 7) Тестване в копие на производствената среда – използването на тестова среда може да доведе до провал на внедряването в реалната производствена среда, тъй като е възможно те да се различават по някакъв начин. Изграждането обаче, на копие на производствената среда може да е скъпоструващо. Вместо това пред-производствената среда ще е изградена като мащабируема версия на реалната производствена среда, което води както до намаляване на разходите, така и до запазване на използвания технологичен стек и нюанси.
- 8) Да могат лесно да се вземат най-актуалните артефакти – ако резултатите са налични винаги за заинтересованите лица и тестерите, това може да намали много количеството преработки, необходими за компилиране на новите версии. В допълнение по-ранното тестване намалява шансовете на дефектите да оцелеят до внедряването. Ранното откриване на грешките в някои случаи намалява количеството работа, необходима за отстраняването им.
- 9) Всеки да може да види резултатите от последната компилация – ще е лесно да бъдат намерени без значение дали компилацията е успешна или не, така че да може да бъдат направени необходимите промени.
- 10) Автоматизирано внедряване - повечето CI системи позволяват стартирането на скриптове, след като завършва компилацията. В повечето случаи е възможно да се напише скрипт за внедряване на прилагането на тестовия сървър, така че всеки да може да го погледнете. По-нататъшно подобрене в начина на внедряване може да се реализира като се направи така, че софтуера да бъде внедрен автоматично на сървъра, често съпроводено от допълнително автоматизиране, за да се предотвратят дефектите и загубата на функционалности.

Автоматизирано управление на кода

Автоматизираното управление на кода, известно също като контрол на версиите и контрол на кода (и е един от аспектите на управлението на софтуерната конфигурация или SCM), е управлението на промените в документи, програми и друга информация, която се съхранява като компютърни файлове. Най-често се използва при разработката на софтуер, където екип от хора могат да се променят едни и същи файлове. Към всяка версия е асоцииран кодов номер, наречен "номер на ревизията". Всяка ревизия е свързана с момент от времето и лицето, което е направило промяната. Ревизиите могат да бъдат сравнявани, възстановени, а някои типове файлове дори и обединявани.

Системите за контрол на версиите (VCS) най-често работят като самостоятелни приложения, но също така са вградени в някои видове софтуер. Софтуерните инструменти за контрол на версиите са от съществено значение за организацията на проектите с много разработчици.



В софтуерната разработка, контрола на версиите е практика, която следи и осигурява контрол над промени в изходния код. Разработчиците на софтуер понякога използват софтуера за контрол на версиите, за да съхраняват документация и конфигурационни файлове, както и изходен код.

Докато екипа проектира, разработва и внедрява софтуера, се инсталират различни версии на един и същи софтуер на различни локации, като и програмистите работят едновременно по обновяванията. Грешки или промени на софтуера често са налични само в определени версии (поради коригирането на грешки и появата на нови с еволюционното развитие на софтуера). Ето защо, за целите на локализирането и отстраняването на грешки е жизнено важно да може да се изтеглят и инсталират различни версии на софтуера, за да се определи в коя версия възниква проблемът. Също така може и да е необходимо да се разработват паралелно две версии на софтуера (например, ако една от версиите има коригирани грешки, но не и нови характеристики (клон), а другата версия е мястото, където се разработват новите характеристики).

В най-елементарно ниво, програмистите биха могли просто да запазят различни копия на програмата и да ги обозначават по подходящ начин. Този прост подход е бил използван при много големи софтуерни проекти. Макар че този метод може да работи, той е неефективен, когато ще се поддържат много, почти идентични копия. Това изисква много самодисциплина от страна на разработчиците и често води до грешки. Като резултат са били разработени системите за автоматизиране на някои или всички от процеса по управление на версиите.

Освен това, в разработката на софтуер, правната и бизнес практики и други среди, става все по-често за един-единствен документ или фрагмент от код, да бъдат редактирани от екип, членовете, на който могат да бъдат географски разпръснати и могат да преследват различни и дори противоречащи интереси. Сложните системи за контрол на версиите, който следи и съхранява информацията за източника на промени в документи и кода могат да бъдат изключително полезни или дори необходими в такива ситуации.



Управление на дистрибуциите

Процесът по управление на дистрибуциите е сравнително нова, но бързо развиваща се дисциплина в рамките на софтуерното инженерство.

Тъй като софтуерни системи, разработката и софтуерните процеси и ресурси стават все по-разпределени, те неизменно стават по-специализирани и по-сложни. Освен това, софтуерните продукти (особено уеб приложения) обикновено са в непрекъснат цикъл на разработка, тестване и внедряване. Добавете към това и еволюцията и нарастващата сложност на платформите, на които тези системи работят, и става ясно, че има много съставни части, които ще работят без проблеми заедно, за да се гарантира успеха и дълготрайната устойчивост на продукта или проекта.

Следователно, съществува нужда от посвещаването на ресурси, които да наблюдават тази интеграция и поток на разработка, тестване, внедряване и поддръжка на тези системи. Въпреки че мениджърите на проекти са правили това в миналото, те обикновено са по-загрижени за аспектите на високо ниво в даден проект, и така често не разполагат с време, за да наблюдават някои от по-техническите или ежедневни аспекти. Управлението на дистрибуциите отговаря на тази необходимост. Те ще имат общи познания на всеки аспект от процеса на разработка на софтуер, различни приложими операционни системи и приложен софтуер или платформи, както и различни бизнес функции и перспективи.

Някои от техниките за управление на дистрибуциите включват управление на:

- Софтуерните дефекти;



- Проблемите;
- Риска;
- Исканията за промяна;
- Исканията за разработка на нови функционалности (допълнителни характеристики);
- Внедряване и пакетиране;
- Нови задачи за разработка.

5.3.9. Описание на инструментите, които ще се ползват от изпълнителя за извършване на разработка, тестване и внедряване

Инструмент	Описание
JetBrains IntelliJ Idea	Среда за разработка
Microsoft Visual Studio	Инструмент за моделиране и изработване на схеми и диаграми
Sparx Systems Enterprise Architect	Специализиран софтуер за моделиране и управление на изисквания: <ul style="list-style-type: none"> - Business Process Modeler; - UML Modeler; - Управление на изискванията.
Microsoft Office Word	Редактор за документи
Microsoft Office Visio	Редактор на векторна графика – създаване на диаграми и схеми за визуализация
Microsoft Office Excel	Електронни таблици
Microsoft Office SharePoint Server	Портал за екипна работа, съхранение и управление на версии на документи
Subversion	Система за контрол на версиите
Oracle Hudson	Специализиран софтуер за създаване на версия на системата (Builds Generator)
Selenium	Автоматизирано тестване на потребителския интерфейс
TestLink	Софтуерна платформа за управление на тестването на софтуерни продукти. Системата предоставя възможност за управление на тестови случаи (test cases), тестови пакети (test suites), тестови планове (test plans), тестови проекти (test projects), а също и разнообразни отчети и статистики.
Atlassian Jira	Възлагане и отчитане на задачи по разработка, тестване и внедряване.

5.3.10. Адаптиране на подхода за софтуерна разработка

Предвиждаме да бъде конфигурирана и адаптирана ИС за корабни инспекции.

Конфигурирането ще протече в следните итерации:

- 1) Итерация 1:
 - a. Рамка на системата;
 - b. Конфигуриране на базова функционалност на ИС за корабни инспекции;
 - c. Конфигуриране на връзка с румънския компонент;
- 2) Итерация 2:
 - a. Интерфейси за съществуващи системи;
 - b. Администриране;
 - c. Връзка с румънския компонент;
- 3) Итерация 3:
 - a. Оптимизиране и изчистване на дефекти;
- 4) Итерация 4:
 - a. Адаптиране на ИС за корабни инспекции
 - b. Оптимизиране и изчистване на дефекти.

5.4. Подход за тестване

Методологии и добри практики: International Software Testing and Qualifications Board

Разработка на тестови сценарии за проверка на реализацията на системата;

Тестване на разработената функционалност по съответния компонент;

Документиране на резултатите от проведените тестове.

Основни средства и техники: Разнообразни видове тестове

Софтуер: Selenium



Jira

Източници: <https://www.istqb.org>

Обосновка: Ще бъдат приложени утвърдени практики на водещи световни професионални организации в областта на софтуерното тестване. Тестването ще бъде планирано и ще бъдат разработени тестови сценарии, което гарантира покритието на продукта с тестове. Ще бъдат приложени различни видове тестове, чрез които ще се тестват различни аспекти на разработката.

5.4.1. Стратегия за тестване

Тестването е етап на изпълнение, по време на който разработения продукт се проверява за съответствие с техническата и функционална спецификация на проекта. Тестването се съотнася с процеса по контрол




на качеството. Тестването се изпълнява паралелно или след процеса по разработка. Задължителен вход за процеса по тестване са резултатите от процеса по разработка. Тестването е тясно свързано с процеса по осигуряване на качеството и се изпълнява едновременно/интегрирано с процесите по анализ, планиране, проектиране, разработване и внедряване. Това осигурява превантивното откриване и ранно отстраняване на грешки.

Тестването ще бъде проведено в три фази:

- Фаза 1: Първичен тест – включва проверка на разработените модули и функционалности, изпълнени върху тестова подсистема (среда);
- Фаза 2: Вторичен тест – включва проверка на разработените модули и функционалности, изпълнени върху продукционна и резервна подсистема, след успешно приключване на първичния тест ;
- Фаза 3: Тестове за натоварване и надеждност – включва тестове на Системата, които гарантират кохерентната работа на съвкупността от компоненти като тестове за натоварване, тестове за непрекъснатост на работата, тестове за възстановяване на предишна стабилна работеща версия на Системата;

И ще се изпълняват за всеки от етапите на изпълнение на поръчката.

Стратегията за тестване се състои от серия различни тестове, които ще проверят работоспособността на системата, включвайки всички нейни компоненти. Основната цел на тези тестове е да разкрие ограниченията на системата и да измери възможностите и.



Тестването започва след етапа на разработка на тестваема част от завършени функционални изисквания за съответния и компонент и следва да се планира като дейност, която е част от общото изпълнение на проекта и преди приемането на Системата. Процесът по тестване ще бъде итеративен и интегриран с цикъла за софтуерна разработка. Тестовите ще бъдат планирани на базата на одобрените изисквания.

Ще бъде прилагана методологията, която разчита на повтарящи се много къси цикли за разработка (итерации). В итеративния подход се приоритизира реализацията на базата на риска, нуждите на крайните потребители и важноста за архитектурата, след което се преминава към по-малките компоненти на системата. Този подход ще се използва през целия последващ цикъл на тестване и внедряване. По този начин тестовия екип получава модули за тестване постепенно и има достатъчно време за тестване преди приключването на проекта. Всяка версия дава възможност за оценка на качеството на крайния продукт. Това също така дава възможност на екипа по разработка да извършва малки промени, когато разходите за прилагането им са ниски. Това намалява риска и увеличава качеството.

Планирането и подготовката на тестовите се извършва преди разработката, веднага след поставянето на изискванията. Още във фазата на проектиране за всички функционални и някои нефункционални изисквания се определят тестовите сценарии, с които те трябва да бъдат покрити. Изготвя се дизайн на това как ще бъде тестван продукта и неговите компоненти. Тестовите се разработват преди да бъдат разработени самите компоненти. При планирането на тестовите се прилага принципа на Парето известен още като „правило 80/20“. Тестовите трябва да се приоритизират и фокуса да бъде върху най-приоритетните причини, които пораждаат множеството проблеми. Тестовите на функционалните и нефункционалните изисквания се групират в тестови сценарии. Изпълнението на тестовите сценарии може да бъде автоматизирано или неавтоматизирано. Решение какъв тест да бъде извършен се взема в етапа на планиране на тестването, като решението е базирано на технологичните особености на конкретната функционалност. В хода на самата разработка процента на реализираните и преминалите

тестове постепенно нараства. В края на итерацията тестовете за итерацията трябва да са проведени и да са преминали успешно. В края на разработката абсолютно всички тестове трябва да са проведени и да са преминали успешно.

Възприети са различни видове тестове на софтуерните продукти. В зависимост от конкретния случай се използват различни техники за тестване. Техниките на тестване са в съответствие с избраните технологии, използвани за разработка. Изборът на това кой конкретен вид тест да бъде използван зависи от конкретния казус (модул, функционалност и др.). Резултатите от тестовете и докладите за дефекти се регистрират, за да може да се извършва контролна функция.

Приемането от Възложителя ще се извършва, при приключване на разработката на пълната функционалност на всички компоненти на Системата, описани в спецификацията на софтуерните изисквания от изпълнявания етап на поръчката. Резултатите от изпълнението ще бъдат нанесени в съответния протокол.

Списък с планираните видове тестове и кратко обяснение за тях следва по-долу.

5.4.2. Начин на провеждане на тестовете

При разработката се извършват следните дейности, свързани с тестването:

- Разработка на тестови сценарии за проверка на реализацията на системата;
- Тестване на разработената функционалност по съответния компонент;
- Документиране на резултатите от проведените тестове.

Изпълнението на дейностите по разработка трябва да бъдат планирани, както по елементи, така и по дефинирани периодични итерации, включващи разработка и тестване на отделните елементи и компоненти. Тестване и представяне на резултатите на Възложителя ще се извършва за всяка итерация, в оперативен порядък. В случай, че Възложителят има изисквания за корекции по реализираната функционалност, те ще бъдат отразени от Изпълнителя до представяне на следващата итерация по разработка на съответния компонент. На база извършената разработка ще бъде извършена интеграцията и тестване в цялостното технологично решение на системата.

5.4.3. Видове тестове

В хода на тестовия процес ще бъдат използвани следните видове тестове, за функционални, нефункционални и приемателни тестове :

Тестове за проверка на изискванията

Това е преглед на спецификацията на изискванията. Целта е да се установи дали те отговарят на изискванията на Възложителя, описани в техническата спецификация и дефинирани в обхвата на проекта. Прегледа на изискванията се извършва от екипа по осигуряване на качеството при изготвяне на спецификацията на изискванията и се одобрява от Възложителя при предаването и.

Тестове за проверка на архитектурата

Архитектурата е техническата концепция за реализацията на изискванията. Може да се опише и като проект на системата на по-високо ниво. Архитектурата ще бъде тествана за съответствие със заданието и цялостните изисквания за архитектурата на Системата.

Функционално тестване

С провеждането на функционалните тестове се цели да се съпоставят реалните действия и състояние на системата с тези описани във функционалната спецификация. Целите на тези тестове са да се провери правилното приемане, обработка и извличане на данни, и правилното осъществяване на бизнес процесите. При наличие на разминавания с функционалната спецификация следва да се направят

предложения и препоръки за тяхното отстраняване. Този вид тестване ще се провежда за всяка фаза, за всеки етап и компонент от разработката на системата.

Системно тестване

Системното тестване започва след като всички функционални и технически изисквания са имплементирани. Целта е да бъдат проверени всички функционални елементи на системата и да гарантира, че системата изпълнява описаните бизнес процеси в съответствие с функционални и технически изисквания. Този начин на тестване позволява да се проследи поведението на цялата системата и всичките и компоненти и дали тя отговаря на очакванията на Възложителя. Този вид тест ще се проведе за Фаза1: Първичен тест и Фаза 2: Вторичен тест.

Интеграционно тестване

Интеграционните тестове ще бъдат изпълнявани, за да се гарантира, че компонентите в изпълнението на системите работят коректно след като са комбинирани, за да изпълняват определена функционалност, както и възможността за обмен на информация с други информационни системи, които са в обхвата на проекта. Целта е компонентите или множествата от компоненти да бъдат проверени за незавършеност или грешки при реализацията. Най-често това са грешки в интерфейсите между компонентите. Този вид тест ще се проведе за Фаза1: Първичен тест.

Тестове за натоварване и производителност

Тестовите за натоварване и производителност целят да докажат приемлива производителност (съгласно заложените параметри за производителност в спецификацията) в смисъла на изпълнение и време за отговор на компонента, като се има предвид очаквания брой потребители на всеки един от компонентите на системата. Целта на теста е да се проверят определени функционални изисквания или бизнес процеси при предварително определени условия, като се проследят данните за производителността на компонента, чрез симулация на очаквания брой потребители на системата, които да я натоварят. По време на тестовите ще бъде измерено времето за отговор на компонента. Този вид тест ще се проведе за Фаза 3: Тестове за натоварване и надеждност.

Тестове за сигурност

Тестовите за сигурност ще определят колко е сигурна новата система. Тестовите трябва да докажат, че е възпрепятстван неототоризиран достъп до поверителна информация, промяна или унищожаване на данните.

Бета тестове

Целта на бета тестовите е да се провери дали софтуерът е готов и може да бъде използван от крайните потребители за изпълнение на техните функции и задачи, за което продуктът е създаден. Бета тестовите ще започнат на етап, при които софтуера е функционално напълно завършен, но все още има вероятност да има неточности. Целта е крайните потребители да имат възможност да проверят дали той работи коректно спрямо функционалните спецификации на мястото, на което работят и в средата, където се очаква да работи системата. Основна роля при този вид тестване имат потребителите на системата. Бета тестовите за Фаза1: Първичен тест ще се проведат в предварително подготвена тестова подсистема, а бета тестовите за Фаза 2: Вторичен тест ще се проведат на продукционната и резервната подсистеми.

Тестове за използваемост

Тестовите за използваемост ще определят колко лесно и интуитивно е използването на разработената система от гледна точка на потребителски интерфейс и с каква лекота биха могли крайните потребители да постигнат специфичните си цели. Тестовите за използваемост включват и оценка за ефективността и бързината, с която потребителите биха могли да завършат задачите си в приложението, след като са запознати с начина му на ползване. Основна роля при този вид тестване имат потребителите на системата. Този вид тест ще се проведе за Фаза 1: Първичен тест.




Тестове за възстановяване

Тестовите за възстановяване целят да проверят колко бързо и ефективно може да се възстанови работа при:

- авария на продукционната подсистема, в които се налага прехвърляне на обслужването на потребителски заявки към резервната подсистема и обратно;
- загуба на комуникация м/у продукционната и резервната подсистема;
- възстановяване от архив;
- възстановяване на предишна стабилна работеща версия на Системата.

Този вид тест ще се проведе за Фаза 3: Тестове за натоварване и надеждност.

Приемни тестове



Приемните тестове ще бъдат последното тестово действие преди внедряване на системата. Целта е да се изгради доверие в софтуера, да се оцени дали системата е готова за внедряване и да се удостовери, че за всички бизнес правила има реализирана функционалност и тя отговаря на функционалната спецификация. Приемните тестове ще се проведат след успешното протичане на трите фази от тестването Фаза1: Първичен тест, Фаза 2: Вторичен тест и Фаза 3: Тестове за натоварване и надеждност за всеки един от етапите на изпълнението на поръчката.

За провеждането им ще бъдат подготвени предварително тестови сценарии, ще се провеждат с участието на Възложител и Изпълнител и ще завършват с подписване на протокол.

5.4.4. Входни и изходни критерии

Критерии за започване на тестовия процес

Критерий за започване на тестовия процес е всички необходими документи по проекта да са изготвени от Изпълнителя и одобрени от Възложителя.

Критерии за преустановяване на тестването

Ако се открият дефекти, които сериозно се отразяват на напредъка на тест, Ръководител тестване може да избере да спре тестването. Критерии, които ще оправдаят преустановяване на тестването са:

- Хардуера/софтуера, необходими за тестването не са налични;
- Разработената функционалност има един или повече критични дефекти (бъгове), които сериозно възпрепятства или ограничава тест прогреса;
- Ресурсите определени за тестване не са на разположение, когато са необходими на тестерите;
- Постъпило е искане за промяна, което засяга функционалностите, които са застъпени в настоящата итерация.



Критерии за подновяване на тестването

Ако тестването е преустановено, подновяването му ще се случи, след като причините предизвикали спирането са отстранени. Когато критичен дефект е причината за спирането, отстраняването му трябва да бъде проверено.

Критерии за спиране на тестовия процес


При покриването на следните критерии се счита, че тестовия процес е преминал успешно:

- ✓ 100% от планираните тестови сценарии трябва да са изпълнени;
- 

- 
- ✓ 100% от всички документирани критични програмни грешки и дефекти трябва да бъдат затворени;
 - ✓ 100% от всички документирани програмни грешки и дефекти с високо въздействие трябва да бъдат затворени;
 - ✓ Фаза 1 : Първичен тест е изпълнен успешно;
 - ✓ Фаза 2 : Вторичен тест е изпълнен успешно;
 - ✓ Фаза 3: Тестове за натоварване и надеждност са изпълнени успешно;
- 


Критерии за приемане

Системата и нейните компоненти ще се приемат в експлоатация когато:

- ✓ Отговарят на изискванията на настоящата техническа спецификация и на утвърдената от Възложителя методология за изпълнение на дейностите;
 - ✓ 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 логически грешки;
 - ✓ Няма грешки или съществен спад на производителността, проявяващи се по време на претоварване, увеличаване на капацитета на базата от данни или автоматични действия по архивиране, индексирание 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);
 - ✓ Съществуват показания, че информационната система може да обработи големи обеми данни без значителен спад на производителността;
 - ✓ Произвежда резултат в очакваното време за отговор;
 - ✓ Успешно са изпълнени Фаза 1 : Първичен тест, Фаза 2 : Вторичен тест и Фаза 3: Тестове за натоварване и надеждност.
- 

5.4.5. Докладване на дефекти и несъответствия

Когато са открити дефекти, тестерите ги отразяват в система за проследяване на дефекти като използват следните атрибути:

- Заглавие – кратко и ясно описание на проблема;
 - Стъпки за възпроизвеждане – описание на пътя за проявяване на проблема;
- 

- Резултат – действителния резултат, получен при изпълнението на тестовия сценарий;
- Очакван резултат – резултатът, който е съобразен със спецификацията на продукта;
- Въздействие – въздействието на проблема върху работоспособността на системата;
- Среда – средата (подсистемата), на която е открит проблема, ако е необходимо;
- Прикачени файлове (снимка, видео, файл) – ако е необходимо.

Ръководителят на екипа следи и анализира вписаните дефекти и по негова преценка ги обработва по един от следните начини:

- Коригирането му се възлага като задача за изпълнение;
- Връща го за допълнително изследване към екипа по осигуряване на качеството като посочва причината за връщането. В този случай статусът се променя на "Изчакващ";
- Приема, че не е дефект (ако като дефект е вписано нормално поведение на системата).

Коригирането на дефекта се възлага като обособена задача и управлението и се осъществява на база правилата за управление на задачи и статусът на дефекта се променя на "В процес на поправяне". След завършване на задачата статусът на дефекта се променя на "Разрешен". Тестер задължително го верифицира и ако установи, че дефектът е отстранен го маркира като „Затворен“ (т.е. отпада от списъка на текущите дефекти).

Система за проследяване на дефекти трябва да е достъпна до тестери, разработчици и всички членове на екипа по проекта.

Дефектите се регистрират в специализирана система за управление на дефекти и несъответствия, базирана на Atlassian Jira.

5.4.6. Инструменти за провеждане на тестовете

Selenium	Автоматизирано тестване на потребителския интерфейс
TestLink	Софтуерна платформа за управление на тестването на софтуерни продукти. Системата предоставя възможност за управление на тестови случаи (test cases), тестови пакети (test suites), тестови планове (test plans), тестови проекти (test projects), а също и разнообразни отчети и статистики.
Atlassian Jira / Team Foundation Server	Възлагане и отчитане на задачи по разработка, тестване и внедряване; Управление на дефекти и несъответствия.

5.4.7. Адаптиране на подхода за тестване

За изпълнение на тестването ще бъде разработен тестови клиент. Ще се използва и олекотен клиент и симулатор за кораби, предоставени от Възложителя. Ще бъде поставен фокус върху тестването на интеграция с други системи.

Поради минималния потребителски интерфейс, тестове за използваемост няма да се използват.



5.5. Подход за внедряване

5.5.1. Общи положения

Внедряването на софтуер е процес, обхващащ всички дейности, които са необходими за да се направи софтуерния продукт достъпен за използване. Внедряването обобщава финалните задачи в обичайния процес по софтуерна разработка, включително миграция на данни, тестване, прехвърляне към новата система и потребителски обучения. Сравнено с традиционните методи, целия процес е компресиран. Като резултат, новата система е изградена, доставена и пусната в действие по-бързо. Задачите на тази фаза са миграция на данни, тестване на цялата система, преход към новата система и потребителски обучения.



Целта на фазата за Внедряване е да направи софтуера достъпен за неговите потребители. Фазата на внедряване може да премине през няколко итерации и включва тестване на продукта, подготовка за нови версии, корекции базирани на обратната връзка от потребителите. На този етап от жизнения цикъл обратната връзка от потребителите трябва да е фокусирана основно върху прецизни настройки, конфигуриране, инсталиране и използваемост. Всички по-съществени структурни въпроси трябва да бъдат решени в по-ранните фази на проекта.


Като цяло процесът за внедряване се състои от няколко взаимосвързани дейности, както и възможните преходи между тях. Тези дейности могат да се извършват в офисите на възложителя, на изпълнителя или и на двете места.

Тъй като всяка софтуерна система е уникална не могат да се дефинират точни процеси или процедури за внедряване. Поради това внедряването трябва да се тълкува като общ процес, който трябва да бъде персонализиран според специфичните изисквания или особености.

5.5.2. Стъпки за внедряване

Ще бъде приложен процеса по внедряване, заложен в Rational Unified Process, според който се извършват следните стъпки:

- 
- 1) Подготовка:
 - 1.1) Предварителен анализ на готовността за внедряване
 - 1.2) Предварително събиране на необходимата за внедряването изходна информация от Клиента.
 - 1.3) Планиране на внедряването
 - 1.4) Съгласуване на плана за внедряване
 - 1.5) Предварително запознаване на Възложителя и страните с ангажиментите им по внедряването
 - 1.6) Разработване на подробно постъпково ръководство за внедряване
 - 1.7) Обучение на екипа от внедрители
 - 1.8) Изготвяне на материали за обучения
 - 1.9) Изготвяне на документация
 - 1.10) Разработване на тестови сценарии за интеграционни и приемни тестове
 - 1.11) Разработване и съгласуване на процедури за поддръжка
 - 1.12) Организиране на help-desk
 - 2) Изпълнение на внедряването:
 - 2.1) Инсталиране и конфигуриране на техническата инфраструктура
 - 2.2) Инсталиране и конфигуриране на базовия софтуер
 - 2.3) Инсталиране и конфигуриране на подсистемите
 - 2.4) Въвеждане на начални данни
 - 2.5) Конфигуриране на интеграционните интерфейси
- 


- 
- 2.6) Провеждане на интеграционни тестове
 - 2.7) Оптимизиране на системата
 - 2.8) Провеждане на приемни тестове
 - 2.9) Обучение на потребителите и администраторите
 - 2.10) Стартиране на работа
- 3) Последващи действия:
- 3.1) Подпомагане на работата със системата
 - 3.2) Поддържане на help-desk за внедрителските екипи
 - 3.3) Управление на версиите на подсистемите

При стартирането на работа със системата, екипи на Изпълнителя ще подпомагат работата със системата:

- На място;
- Чрез портал за подпомагане на потребителите;
- Чрез гореща телефонна линия за подпомагане на потребителите.

Инсталациите и конфигурациите ще бъдат извършени с участие на администратори от страна на Възложителя така, че да бъде извършено предаване на знания.


Допълнително ще бъдат обучени ключовите потребители на системата. Те ще имат осигурен достъп до тестова среда така, че да придобият увереност за работа със системата преди въвеждането и в експлоатация.



5.6. Механизъм и процедури за включване на граждани в процесите по разработка, тестване и идентифициране на пропуски на софтуера

Изпълнителят ще извърши анализ на възможностите за включване на граждани в процесите по разработка, тестване и идентифициране на пропуски на софтуера. Този анализ ще бъде изпълнен по начин, даващ възможност за активно участие от страна на гражданите и събиране на информация с цел обратна връзка за функционалната реализация на разработеното приложение. Като предварителна визия предлагаме да бъдат използвани следните механизъм и процедури за реализирането на такива процеси:



- 
- Да бъдат публикувани за достъп от граждани спецификациите и ръководствата, резултат от изпълнението на поръчката (по преценка на Възложителя).
 - Да бъде публикувана в Интернет точка за достъп до тестова инстанция на интеграционния интерфейс на адаптера (по преценка на Възложителя).
 - Изпълнителят ще предаде пълният изходен код на разработката. По преценка на Възложителя, той може да бъде публикуван в хранилището, поддържано от Държавна агенция „Електронно управление“.
 - Ще бъде осигурен канал за получаване на обратна връзка от граждани чрез специален адрес на електронна поща.

6. Документация

6.1. Изисквания към документацията

Цялата документация и всички технически описания, ръководства за работа, администриране и поддръжка на Системата, включително и на нейните съставни части, ще бъдат налични и на български език.

Всички документи ще бъдат предоставени от Изпълнителя в електронен формат (ODF/ /Office Open XML/MS Word DOC/RTF/PDF/HTML или др.), позволяващ пълнотекстово търсене/търсене по ключови думи и копиране на части от съдържанието от оригиналните документи във външни документи, за вътрешна употреба на възложителя;

Навсякъде, където в документацията има включени диаграми или графики, те ще бъдат вградени в документите в оригиналния си векторен формат;

Изпълнителят ще изготви и предостави на Възложителя:

- Детайлна техническа документация на програмния приложен интерфейс (API), включително за поддържаните уебслужби, команди, структури от данни и др.
- Детайлна техническа документация за схемата на базата данни – структури за данни, индекси, дялове, съхранени процедури, конфигурации за репликация на данни и др.
- Ръководства на потребителя и администратора за работа и администриране на Системата;
- Обща информация, инструкции и процедури за администриране и поддръжка на приложните сървъри, сървърите за бази данни и др.
- Обща информация, инструкции и процедури за администриране, архивиране и възстановяване, и поддръжка на сървъра за управление на бази данни.
- Детайлна техническа документация за схемата на базата данни – структури за данни, индекси, дялове, съхранени процедури, конфигурации за репликация на данни и др.

Цялата документация по изпълнението ще отговаря на определени стандарти и утвърдени добри практики по отношение на оформление, структура, контрол на версиите, критерии за проследимост и контрол на качеството на документите. Изпълнителят ще предложи стандарти за подготовка на документация, съобразени с предлаганата от него проектна методология.

6.2. Прозрачност и отчетност

Всички документи ще се изготвят на български език и се предоставят в един хартиен екземпляр и на електронен носител.

6.3. Спецификация на софтуерните изисквания

Изпълнителят на настоящата поръчка ще дефинира в детайли конкретния обхват на реализация на софтуерната разработка и ще документира изискванията към софтуера в детайлна техническа спецификация (спецификация на софтуерните изисквания), която ще послужи за пряка изходна база за разработка.

При документирането на изискванията, с цел постигане на яснота и стандартизация на документите, ще се използва утвърдена нотация за описание на бизнес модели. Изготвената детайлна техническа спецификация (спецификация на софтуерните изисквания) ще бъде представена за одобрение на Възложителя. В случай на забележки, корекции или допълнения от страна на Възложителя Изпълнителят е ще ги отрази в детайлната техническа спецификация (системен проект).

6.4. Техническа документация

При изпълнение на дейностите по настоящата поръчка, изпълнителят ще изготви следната техническа документация, съдържаща:

- Ръководство за инсталиране, конфигуриране и администриране;
- Ръководство за потребителите;
- Детайлно техническо описание на софтуерните модули и базата данни;
- Описание на изходния програмен код

6.5. Протоколи

Изпълнителят ще изготвя протоколи от изпълнението на различните етапи на проекта заедно със съпътстващите ги документи – резултати от изпълнението на етапите.

6.6. Комуникация и доклади

Виж. по-горе т. 3.2 – Управление на комуникацията.

7. Предпоставки за успешно изпълнение на поръчката

За успешното изпълнение на поръчката е необходимо да бъдат съобразени следните предпоставки:

- Ангажираност на Възложителя в проекта – да определи проектен екип и ръководител, за които задачите по проекта ще бъдат с висок приоритет и ръководителят ще има необходимата оторизация.
- В проектния екип на Възложителя да бъдат подбрани експерти с познания в предметната област, които да могат да изпълняват ролята на ключови потребители.

000198

към

ТЕХНИЧЕСКО ПРЕДЛОЖЕНИЕ

за участие в открита процедура за възлагане
на обществена поръчка с предмет:

**„Доставка на интегрирана система за корабна инспекция –
софтуерен пакет с електронна база данни и оборудване“**

Предложение за управление на риска при
изпълнение на поръчката

2018 г.


Съдържание

1. Въведение.....	3
2. Подход за управление на риска	3
2.1. Планиране на управлението на риска	4
2.2. Идентифициране на рисковете	5
2.3. Качествен анализ на риска.....	6
2.4. Количествен анализ на риска	7
2.5. Планиране на ответни действия	8
2.6. Наблюдение и контрол на рисковете	9
3. План за управление на риска.....	10
3.1. Процес.....	10
3.2. Роли и отговорности	10
3.3. Таксономия и категоризиране на рисковете.....	11
3.4. Таксономия на рисковете при софтуерна разработка.....	11
3.5. Регистър на рисковете	13
3.6. Инструменти за управление на риска	15
4. Конкретни рискове	16
4.1. Анализ на конкретни рискове, идентифицирани от Възложителя.....	16
4.2. Анализ на допълнителни, относими към поръчката рискове	24



1. Въведение

В настоящото приложение 2 към Техническото си предложение сме представили:


- Подход за управление на риска, който ще прилагаме при изпълнението на поръчката;
 - Анализ на идентифицираните в Техническата спецификация на Възложителя възможни рискове;
 - Анализ на идентифицирани от нас допълнителни, относими към поръчката рискове;
 - Предпоставки за успешно изпълнение на поръчката.
- 

Рисковете са анализирани по отношение на вероятността от тяхното настъпване, въздействието им върху изпълнението на дейностите и сферата им на влияние. Предложени са мерки и обосновка за предотвратяване или минимизиране на влиянието за всеки един от рисковете. Предложените мерките са аргументирани, приложими и изпълними за премахване/минимизиране на всеки един от посочените рискове.

2. Подход за управление на риска

През времето за изпълнение на проекта Изпълнителят ще следи рисковете, ще оценява тяхното влияние, ще анализира ситуацията и ще идентифицира (евентуално) нови рискове. В хода на изпълнение на поръчката Изпълнителят ще докладва състоянието на рисковете на месечна база.

При управлението на проекта ще бъде използвана практикуваната при нас система от правила, принципи и стандарти на Института за Управление на Проекти (Project Management Institute, PMI), систематизирани в Ръководство за Система от Знания за Управление на Проекти (PMBOK Guide) – пето издание. Това включва и Система за управление на риска, която е представена по-долу.



Рискът се отнася до отклонението от един или повече резултати на едно или повече бъдещи събития от тяхната очаквана стойност. Технически, стойността на тези резултати може да е позитивна или негативна. Положителният риск се разглежда като възможност, а при общата употреба на думата риск се фокусира само върху потенциалната вреда (загуба на позитивни резултати), която може да възникне от бъдещо събитие, което да произтече или от влизане в разноски ("риск от снижение" - на английски: downside risk), или от неспособност да се придобие някаква печалба ("риск на изкачването" - на английски: upside risk). Рискът по дефиниция е събитие, което е извън нашия пряк контрол и може да повлияе отрицателни или положително върху постигането на целите на проекта.

Управлението на риска в проекта включва процесите по планиране, идентификация, анализ, планиране на ответните действия и наблюдение и контрол на рисковете. Целта на управлението на риска в проекта е да се повиши вероятността и влиянието на позитивните събития и да се намали вероятността и влиянието от негативните събития в проекта.

Процесите по управление на риска в проекта включват:

- **Планиране на управлението на риска** – Процесът на определяне как ще се изпълняват дейностите по управление на риска в проекта.
- **Идентифициране на рисковете** – Процесът на определяне кои рискове могат да повлияят на проекта и документиране на техните характеристики.
- **Качествен анализ на риска** – Процесът на приоритизиране на рисковете за целите на бъдещ анализ, както и действията по оценка на вероятността да се случат и влиянието им върху проекта.
- **Количествен анализ на риска** – Процесът на количествено и стойностно анализиране на ефекта върху проекта при случване на риска.
- **Планиране на ответни действия** – Процесът на разработване на опции и действия за повишаване на благоприятните възможности и намаляване на заплахите пред проектните цели.

Handwritten signature

- **Наблюдение и контрол на рисковете** – Процесът на изпълнение на планираните ответни действия, проследяване на идентифицираните рискове, наблюдение на остатъчните рискове, идентифициране на нови рискове и оценка на процесите по управление на риска в проекта.

Описаните процеси си взаимодействат с останалите процеси по управление на проекта. По всеки от процесите работят един или повече хора в зависимост от спецификата на проекта. Всеки от процесите се изпълнява поне веднъж във всеки проект и се появява в една или няколко от фазите на проекта.

Въпреки, че процесите са описани като дискретни елементи с ясно обособен вход и изход, на практика те се допълват и си взаимодействат.

Проектният риск винаги е в бъдещето. Рискът е несигурно събитие или условие, което ако се случи, ще повлияе върху поне една от целите на проекта. Целите включват обхват, график, качество и цена. Рискът може да се случи поради една или множество причини и ако се случи може да повлияе на проекта по един или няколко начина. Причините могат да се състоят в появата на изискване, допускане, ограничение или условие, което създава вероятност за позитивен или негативен резултат.

Ако някое неочаквано събитие се случи, но може да повлияе върху разходите, графика или изпълнението. Потенциални рискове следват и от приетите практики – недостатъчно зрял процес за управление на проекти, липса на интегрирана система за управление, множество конкурентни проекти, зависимости от външни участници, които не могат да бъдат контролирани.

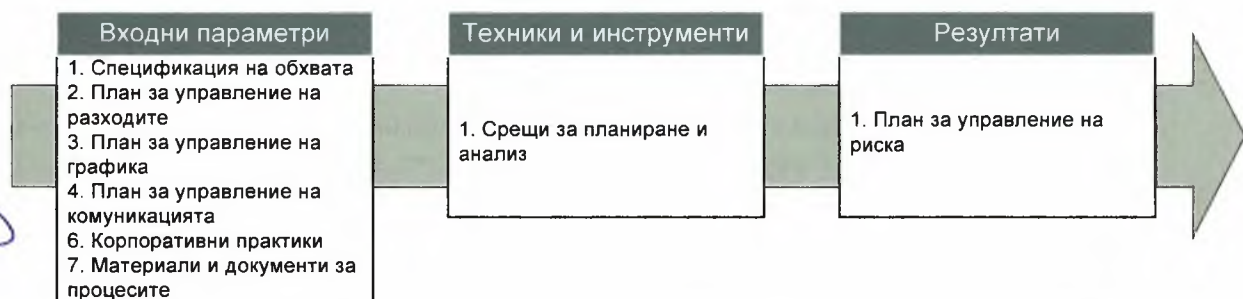
Проектният риск произтича от несигурността, съществуваща във всеки един проект. Известни рискове са тези, които са били идентифицирани и анализирани, което прави възможно да се планират ответни действия.

Организациите възприемат риска като ефект на несигурност на техните проекти или цели. Организациите и мениджмънта са склонни да приемат различна степен на риск. Това се нарича толеранс към риска. Рисковете, които са заплаха към проекта могат да бъдат приети, ако са в този толеранс, и ако и приемането на риска ще донесе достатъчно полза. Например, приемането на график с множество дейности, изпълнявани паралелно е риск, който се приема заради ползата от по-ранна дата на завършване.

За да бъдат успешни, организациите трябва да приемат, че ще управляват риска проактивно и непрекъснато през целия проект. Рискът съществува в момента, в който възниква проекта. Управлението на риска е динамичен процес, който се изпълнява в хода на целия проект. Проектните рискове могат да се променят в хода на проекта, като съществуващи рискове да отпаднат, да бъдат идентифицирани нови рискове, както и да се променя състоянието на съществуващи такива. Списъкът с актуални рискове следва да се актуализира посредством анализ на периодичните отчети по проектите. Напредването на проекта без фокус върху проактивното управление на риска води до повишаване на влиянието, което рискът може да окаже при възникването си и потенциално може да доведе до провал на проекта.

Несъобразяване с кой да е КРИТИЧЕН ФАКТОР ЗА УСПЕХ генерира допълнителен риск.

2.1. Планиране на управлението на риска



Планиране на управлението на риска е процесът на определяне как ще се изпълняват дейностите по управление на риска в проекта. Внимателното и посветено планиране повишава вероятността за успех на останалите процеси за управление на риска. Планирането на процесите за управление на риска е

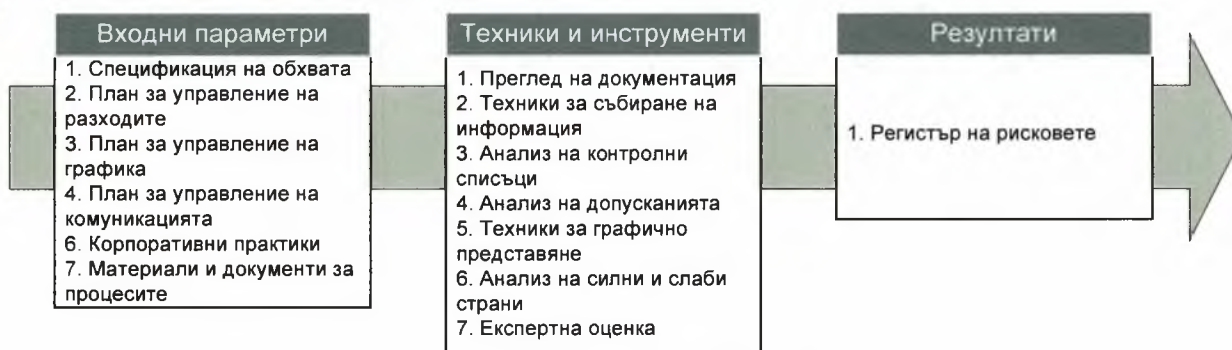
Handwritten signature

важно, за да се осигури, че степента, типа и подхода за управление на риска са адекватни на потенциалните рискове и важността на проекта за съответната организация.

Планирането също е важно за осигуряването на достатъчно ресурси и време за дейностите по управление на риска. Процесът по планиране управлението на риска трябва да стартира с възникването на проекта и трябва да се изпълнява заедно с останалото планиране по проекта.

- **Входни параметри**
 - Спецификация на обхвата
 - План за управление на разходите
 - План за управление на графика
 - План за управление на комуникацията
 - Практики
 - Материали и документи за процесите
- **Техники и инструменти**
 - Среци за планиране и анализ
- **Резултати**
 - План за управление на риска

2.2. Идентифициране на рисковете



Идентифициране на рисковете е процесът на определяне кои рискове могат да повлияят на проекта и документиране на техните характеристики. Участници в процеса на идентификация могат да бъдат: проектния ръководител, членовете на проектния екип, екипът за управление на риска, клиенти, експерти в предметната област, експерти извън проектния екип, други проектни ръководители, заинтересовани лица и експерти по управление на риска.

Въпреки, че въпросните хора са ключови участници в идентификацията на риска, всички членове на екипа се окуражават да участват в идентифицирането на рисковете.

Идентифицирането на рисковете е итеративен процес т.к. през целия цикъл на проекта е възможно да възникват нови рискове или да стават известни съществуващи рискове. Честотата на итерациите и участниците в процеса зависят от конкретната ситуация.

Форматът на описване на рисковете трябва да бъде еднотипен, за се осигури възможността за сравнение на ефекта от риска спрямо този на останалите рискове в проекта. Процесът трябва да включва участие на проектния екип, така че той да развие чувство за отговорност и съпричастност спрямо рисковете и ответните действия. Заинтересованите лица извън проектния екип могат да предоставят допълнителна обективна информация.

- **Входни параметри**
 - Спецификация на обхвата
 - План за управление на разходите

- План за управление на графика
- План за управление на комуникацията
- Корпоративни практики
- Материали и документи за процесите

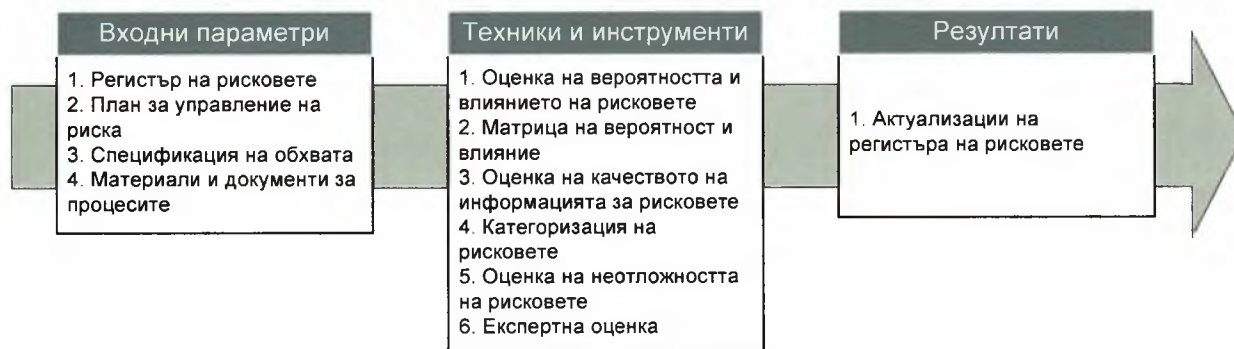
- **Техники и инструменти**

- Преглед на документация
- Техники за събиране на информация
- Анализ на контролни списъци
- Анализ на допусканията
- Техники за графично представяне
- Анализ на силни и слаби страни
- Експертна оценка

- **Резултати**

- Регистър на рисковете

2.3. Качествен анализ на риска



Качествен анализ на риска е процесът на приоритизиране на рисковете за целите на бъдещия анализ, както и действията по оценка на вероятността да се случат и влиянието им върху проекта. Организацията могат да подобрят изпълнението на проекта чрез фокусиране върху рисковете с висок приоритет. При качествения анализ на риска се оценява приоритета на идентифицираните рискове, като се вземе предвид вероятността за случване на риска, степента на влияние върху проектните цели, а също и други фактори като време за реакция, приет от организацията толеранс към риск и приетите ограничения на проекта – цена, график, обхват и качество.

Тези оценки се отразяват върху отношението на проектния екип и другите заинтересовани лица към риска. И т.к. отношението към риска води до отклонение при анализа на идентифицираните рискове, трябва да се обърне специално внимание върху оценката на това отклонение и коригирането му.

Установяването на определени нива на вероятност и влияние може да намали влиянието на отклоненията. Оценката на качеството на наличната информация за проектните рискове също спомага за по-прецизно определяне на важността на риска за проекта.

Качественият анализ на риска трябва да се изпълнява по време на целия жизнен цикъл на проекта за да бъде актуален спрямо промените в проектните рискове. От качествения анализ на риска може да се продължи към количествен анализ или директно към планиране на ответните действия.

- **Входни параметри**

- Регистър на рисковете

- План за управление на риска
- Спецификация на обхвата
- Материали и документи за процесите

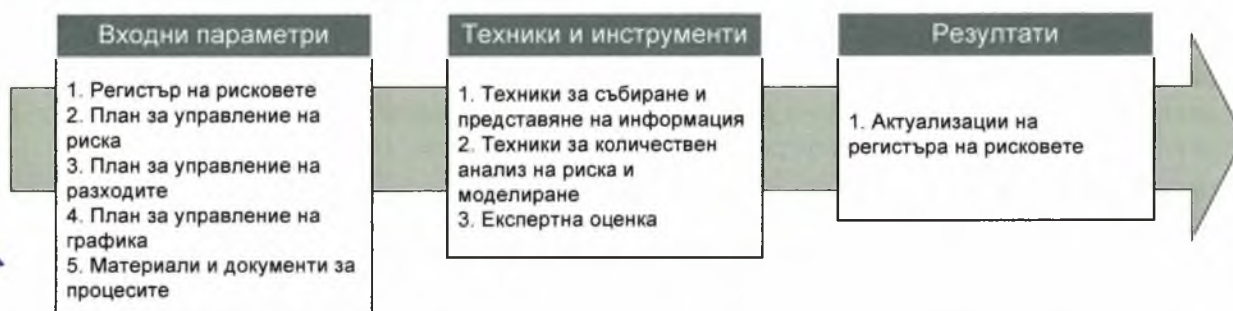
- **Техники и инструменти**

- Оценка на вероятността и влиянието на рисковете
- Матрица на вероятност и влияние
- Оценка на качеството на информацията за рисковете
- Категоризация на рисковете
- Оценка на неотложността на рисковете
- Експертна оценка

- **Резултати**

- Актуализации на регистъра на рисковете

2.4. Количествен анализ на риска



Количествен анализ на риска е процесът на количествено и стойностно анализиране на ефекта върху проекта при случване на риска. Количественият анализ се изпълнява върху рисковете, които са приоритизирани при качествения анализ като значими за успешното изпълнение на проекта. Чрез този процес се дават числови изражения на ефекта от възникване на рисковете – за конкретен риск или общо. Количествената информация дава изходна информация при взимането на решения относно управлението на рисковете.

Изпълнението на количествен анализ обикновено следва процеса на качествения анализ. В някои случаи не се налага количествен анализ за планирането на ответни действия. Наличието на време и бюджет, както и конкретната нужда за качествен и количествен анализ определя подхода във всеки конкретен проект. Количественият анализ трябва да се повтори след планиране на ответните действия, а също така и по време на наблюдението и контрол на рисковете. Тенденциите в проекта показват нуждата от повече или по-малко действия за управление на риска.

- **Входни параметри**

- Регистър на рисковете
- План за управление на риска
- План за управление на разходите
- План за управление на графика
- Материали и документи за процесите

- **Техники и инструменти**

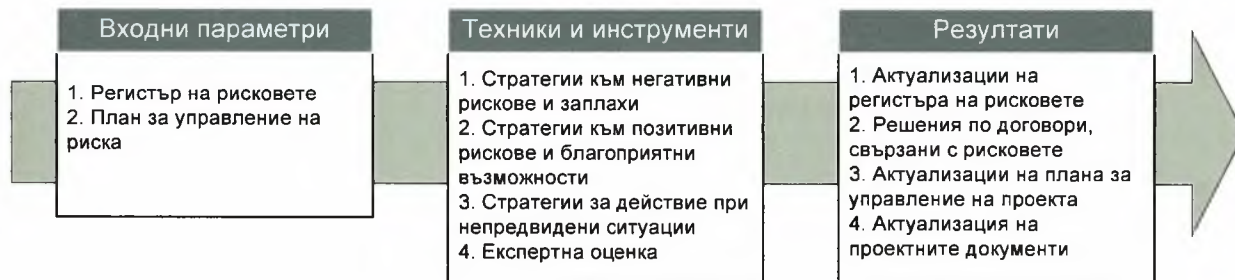
- Техники за събиране и представяне на информация

- Техники за количествен анализ на риска и моделиране
- Експертна оценка

- **Резултати**

- Актуализации на регистъра на рисковете

2.5. Планиране на ответни действия



Планиране на ответни действия е процесът на разработване на опции и действия за повишаване на благоприятните възможности и намаляване на заплахите пред проектните цели. Той следва изпълнението на количествен и качествен анализ. Включва определяне на конкретен човек - отговорник за изпълняване на ответните действия при проява на риска. Планирането на ответни действия се отнася към рисковете в зависимост от техния приоритет, при това се включват съответни дейности в графика, бюджета на проекта и плана за управление.

Планираните ответни действия трябва да бъдат подходящи в зависимост от значимостта на рисковете, ценово ефективни, реалистични в контекста на проекта, съгласувани между заинтересованите страни и с назначен конкретен отговорник. Ответните действия трябва да бъдат навременни.

Използват се следните подходи за ответни действия при негативни рискове: избягване, прехвърляне, смекчаване и приемане. При позитивните рискове подходите са възползване и споделяне.

- **Входни параметри**

- Регистър на рисковете
- План за управление на риска

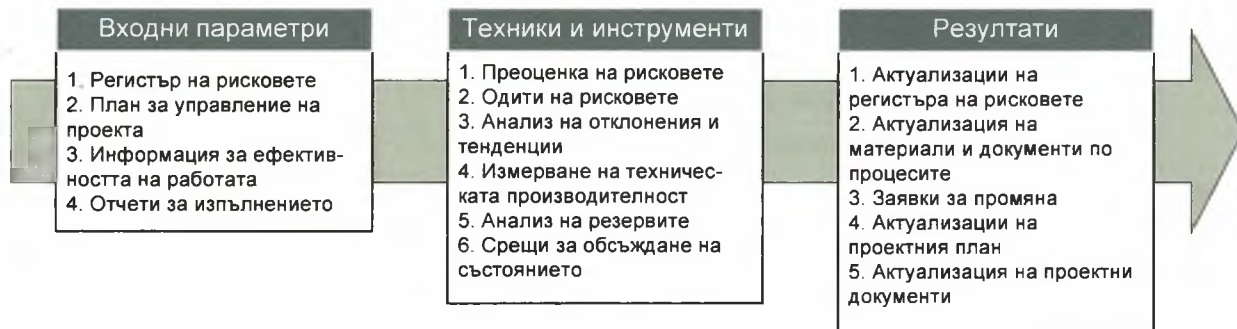
- **Техники и инструменти**

- Стратегии към негативни рискове и заплахи
- Стратегии към позитивни рискове и благоприятни възможности
- Стратегии за действие при непредвидени ситуации
- Експертна оценка

- **Резултати**

- Актуализации на регистъра на рисковете
- Решения по договори, свързани с рисковете
- Актуализации на плана за управление на проекта
- Актуализация на проектните документи

2.6. Наблюдение и контрол на рисковете



Наблюдението и контрола на рисковете е процесът на изпълнение на планираните ответни действия, проследяване на идентифицираните рискове, наблюдение на остатъчните рискове, идентифициране на нови рискове и оценка на процесите по управление на риска в проекта.

Планираните ответни действия, включени в проектния план се изпълняват по време на целия жизнен цикъл на проекта. Проектът трябва да бъде постоянно наблюдаван за нови, променени или отпаднали рискове.

При наблюдението и контрола на рисковете се прилагат техники като анализ на отклонения и тенденции, които изискват използване на информация за изпълнението. Друга цел на процеса е да се определи дали:

- проектните допускания са валидни;
- оценените рискове са се променили или отпаднали;
- се спазват политиките и процедурите за управление на риска;
- финансовите и времеви резерви трябва да се променят в зависимост от текущата оценка на рисковете.

Наблюдението и контролът на рисковете може да включва избор на алтернативни стратегии, изпълнение на резервен план, прилагане на коригиращи действия и промяна на проектния план. Отговорникът за риска периодично докладва на проектния мениджър ефективността на плана, неочаквани ефекти и необходимостта от корекции, необходими за адекватно управление на риска. Процесът също включва актуализацията на материали и документи по процесите, базата данни с „научени уроци“ и шаблони, които да бъдат използвани при бъдещи проекти.

• Входни параметри

- Регистър на рисковете
- План за управление на проекта
- Информация за ефективността на работата
- Отчети за изпълнението

• Техники и инструменти

- Преоценка на рисковете
- Одити на рисковете
- Анализ на отклонения и тенденции
- Измерване на техническата производителност
- Анализ на резервите
- Срещи за обсъждане на състоянието

• Резултати

- Актуализации на регистъра на рисковете
- Актуализация на материали и документи по процесите
- Заявки за промяна
- Актуализации на проектния план
- Актуализация на проектни документи

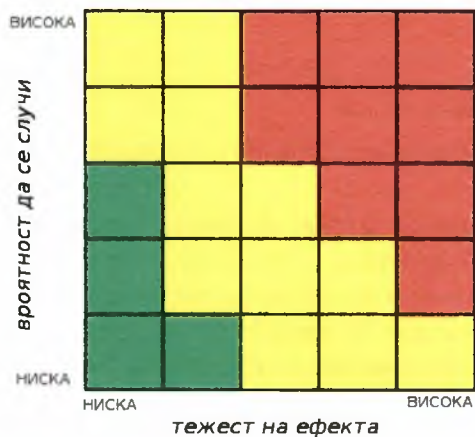
3. План за управление на риска

3.1. Процес

Рискове се идентифицират посредством стандартни методи от теорията на управлението на риска. Веднъж идентифицирани рискове, те трябва да бъдат докладвани посредством периодичен доклад за изпълнението на проект. Също така специфичен риск може да бъде докладван и с отделен доклад, афиширащ специфичен риск.

Общия регистъра на риска се актуализира периодично на база на докладите и най-вече от докладваната информацията за риск, проблеми, текущ статус и препоръки. Самите доклади служат за регистър на проектните рискове.

Всички идентифицирани рискове се вписват в регистъра. Качествен анализ се изпълнява за всички идентифицирани рискове.



ЛЕГЕНДА:

- рискът може да бъде пренебрегнат
- рискът трябва да бъде анализиран
- рискът трябва да бъде управляван

Рисковете с ниска вероятност и влияние върху проекта могат да бъдат пренебрегнати. Рисковете със средна и висока вероятност или среден и висок ефект трябва да бъдат следени и анализирани. За рисковете с висока вероятност и ефект задължително се изпълнява количествен анализ и трябва да бъдат управлявани.

3.2. Роли и отговорности

Отговорен за цялостния процес по управление на риска е проектния ръководител.

В помощ на проектните ръководители работи борд за управление на риска, който включва:

- ръководители на екипи;
- представители на Възложителя;

- експерти в предметната област;
- представител на спонсора на проекта или партньорска организация.

Всеки член на екипа участва в процеса на идентифициране на рисковете. Качествения анализ на риска се извършва от борда за управление на риска. Количественият анализ на риска и планирането на ответни действия се извършва от проектния ръководител, подпомогнат от борда за управление на риска. За изпълнението на количествен анализ и планиране на ответни действия проектният ръководител може да използва външни експерти в специфичната проектна област.

Наблюдението и контролът на рисковете се извършва от борда за управление на риска.

Минимум веднъж на няколко дена, а при необходимост и по-често Проектният ръководител докладва текущото състояние на проектните рискове. Регистърът на рисковете е достъпен за всички заинтересовани лица.

3.3. Таксономия и категоризиране на рисковете

Използват се следните категории при качествения анализ на рисковете:

- Категории за „Вероятност от настъпване“: малка, средна, голяма;
- Категория „Степен на въздействие“: малко, средно, голямо;
- В зависимост от вероятността и въздействието следва приоритетът на риска:

Вероятност Въздействие	Ниска	Несъществена	Умерена	Съществена	Висока
Много голяма (81 - 100%)	Среден приоритет	Среден приоритет	Висок приоритет	Висок приоритет	Висок приоритет
Голяма (61 - 80%)	Среден приоритет	Среден приоритет	Висок приоритет	Висок приоритет	Висок приоритет
Средна (41 - 60%)	Нисък приоритет	Среден приоритет	Среден приоритет	Висок приоритет	Висок приоритет
Ниска (21 - 40%)	Нисък приоритет	Среден приоритет	Среден приоритет	Среден приоритет	Висок приоритет
Минимална (1 - 20%)	Нисък приоритет	Нисък приоритет	Среден приоритет	Среден приоритет	Среден приоритет

При количественият анализ вероятността се оценява в проценти, а влиянието в пари – разход при възникване на риска.

При управлението на риска се използват следните състояния за проследяване на рисковете: постоянен, бъдещ, текущ, затихнал, отминал.

3.4. Таксономия на рисковете при софтуерна разработка

Т. нар. таксономия на рисковете е стандартен начин за класифициране и организиране на рисковете. При изпълнението на проекта ще бъде прилагана таксономия на рисковете при софтуерна разработка на Института по софтуерно инженерство към Университет „Карнеги Мелън“:



Software Engineering Institute, Carnegie Mellon University

Handwritten signature

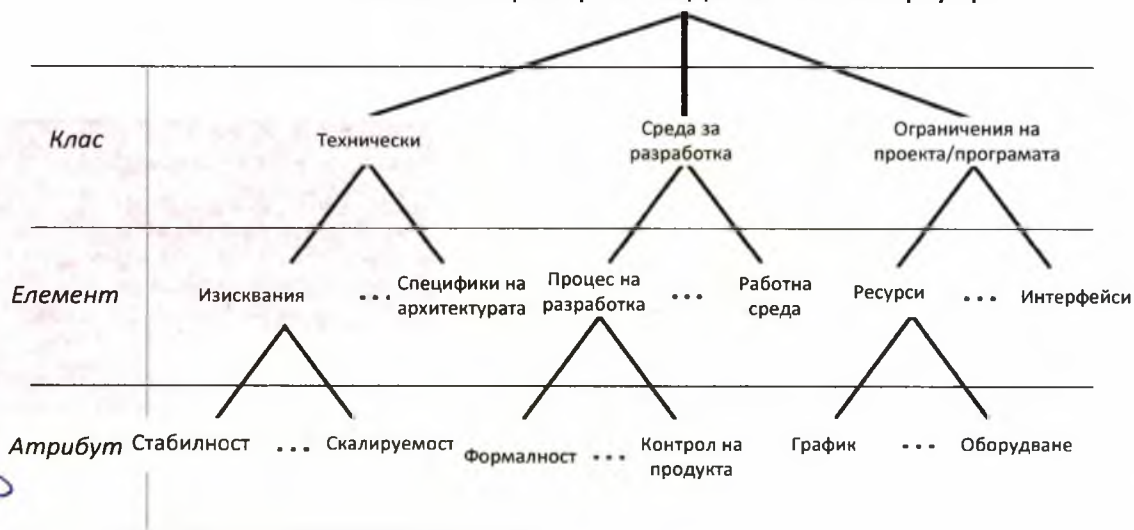
Предложената класификация следва процесите на софтуерното производство. Използването на стандартна класификация, извлечена от опита в различни проекти спомага при идентификацията на рисковете да се минимизират възможностите за пропуск.

Методът за идентификация на рисковете на Института по софтуерно инженерство се базира на следните приемания:

- Рисковете при софтуерното производство обикновено са известни на проектните екипи, но често са лошо комуникирани.
- За последователно управление на рисковете е нужен структуриран и повторяем метод за идентифициране.
- Ефективното идентифициране на рисковете трябва да покрива всички ключови области от разработката и поддръжката на проекта.
- Процесът за идентифициране на рисковете трябва да създаде и поддържа среда за събиране на информация, която да избягва използването на субективна, спорна и неубедителна информация.
- Не трябва да се прави оценка за успеха на проекта на базата на броя и естеството на откритите рискове.

Таксономията на рисковете е организирана на три нива: класове, елементи и атрибути:

Рискове при производството на софтуер



Таксономия на рисковете при софтуерна разработка:

A. Технически	B. Среда за разработка	C. Ограничения на проекта и програмата
1. Изисквания <ol style="list-style-type: none"> Стабилност Пълнота Яснота Валидност Осъществимост Прецедент Скалируемост 2. Дизайн <ol style="list-style-type: none"> Функционалност Леснота Интерфейси Производителност 	1. Процес за разработка <ol style="list-style-type: none"> Формалност Приложимост Контрол на процеса Познаване на процеса Контрол на продукта 2. Инструменти за разработка <ol style="list-style-type: none"> Капацитет Приложимост Използваемост Познаване Надеждност 	1. Ресурси <ol style="list-style-type: none"> График Персонал Бюджет Оборудване 2. Договор <ol style="list-style-type: none"> Тип на договора Ограничения Зависимости 3. Интерфейси <ol style="list-style-type: none"> Клиент Партньори

Handwritten signature

<p>e. Тестваемост</p> <p>f. Хардуерни ограничения</p> <p>g. Невъзможност за разработка</p> <p>3. Код и тестване на модулите (unit test7)</p> <p>a. Изпълнимост</p> <p>b. Тестване</p> <p>c. Кодиране/Реализация</p> <p>4. Интеграция и тестване</p> <p>a. Среда</p> <p>b. Продукт</p> <p>c. Система</p> <p>5. Специфики на архитектурата</p> <p>a. Възможност за поддръжка</p> <p>b. Надеждност</p> <p>c. Безопасност</p> <p>d. Сигурност</p> <p>e. Човешки фактори</p> <p>f. Спецификации</p>	<p>f. Поддръжка</p> <p>g. Възможност за доставяне</p> <p>3. Процес за управление</p> <p>a. Планиране</p> <p>b. Организация на проекта</p> <p>c. Управленски опит</p> <p>d. Интерфейси на програмата</p> <p>4. Методи за управление</p> <p>a. Мониторинг</p> <p>b. Управление на персонала</p> <p>c. Осигуряване на качеството</p> <p>d. Управление на конфигурацията</p> <p>5. Работна среда</p> <p>a. Отношение към качеството</p> <p>b. Кооперативност</p> <p>c. Комуникация</p> <p>d. Морал</p>	<p>c. Подизпълнители</p> <p>d. Основен контрактор</p> <p>e. Управление на компанията</p> <p>f. Доставчици</p> <p>g. Политика</p>
--	--	--

Посочената класификация на рисковете ще бъде разширена с релевантни категории рискове предвид широкия обхват на проекта.

3.5. Регистър на рисковете

Рисковете за проекта ще се следят през целия период на изпълнение. За тази цел ще се поддържа списък на рисковете. Той се подготвя в началото на проекта и се поддържа в актуално състояние по време на целия проект. Преглежда се на всяка среща.

При оценката на всеки един от рисковете е оценена следната задължителна информация по рискове:

1. Обхват и степен на въздействие (последствия) на риска върху изпълнението на обществената поръчка;
2. Мерки за недопускане/предотвратяване на риска, в приложимите случаи, съответно обосновка за невъзможността да се предприемат подобни мерки в конкретния случай;
3. Мерки за преодоляване на последиците при настъпване на риска.

С помощта на този регистър ще може да се проследяват всички потенциални и възникнали рискове по време на изпълнение на проекта, което от своя страна ще позволи навременно извършването на превантивни действия с цел минимизиране на риска или реагиране при възникването на някой от тях.

Списъкът на рисковете представлява структурирано описание на известните и реално стоящи рискове пред проекта, подредени в низходящ ред по значение. Към всеки риск се привързват мерки за ограничаване на последствията или действия при настъпване на риска. Списъкът на рисковете трябва да отразява критичните и сериозни рискове. Изготвя се в следния примерен вид:

Handwritten signature

№	Описание	Последствия и сфера на влияние	Отговорник	Обща експозиция	Степен на значимост	Вероятност от настъпване	Индикатор	Мерки за минимизиране или предотвратяване на негативното въздействие и тяхната обосновка

Рискът се идентифицира с пореден номер, който се записва в първата колона. Втората колона съдържа описание на риска, а третата – резюме на въздействие върху изпълнението на проекта. В колона "Отговорник" се посочва(т) лицето или организацията (лицата или организациите), което отговаря (които отговарят) за преодоляването на съответния риск. Следваща колона е „Вид“ и съответно се определя дали е технически, външни, организационни, управленски. Степен на въздействие – отклонението от планираните графици, усилия и разходи, ако рискът действително се реализира.

Степен на въздействие	Вероятност от настъпване
Висока	Много висока (81-100%)

Степен на въздействие
Изберете стойността, която най-добре описва какво ще е отклонението от планираните графици, усилия и разходи, ако рискът действително се реализира.

Handwritten signature

Вероятност от настъпване (изразява се като процент) – Вероятността рискът действително да се реализира (обикновено се изразява като процент).

Вероятност от настъпване	Индикатор
Много висока (81-100%)	

Вероятност от настъпване
Изберете стойността, която най-добре описва вероятността рискът действително да се реализира.

Handwritten signature

Индикатор - За всеки риск от списъка се идентифицира подлежащо на измерване състояние, настъпването на което означава, че рискът действително се е реализирал. Тези състояния представляват индикатори (лакмуси) за появата на риска (с чиято помощ разбираме, че рискът е вече реалност).

Стратегия за ограничаване - Разработването на планове за ограничаване на риска означава да се намалят последствията от настъпването на риска. За някои рискове се изисква да се опишат и действията, чието изпълнение зависи от настъпването на риска. Тук се разглеждат всички мерки за недопускане/предотвратяване на риска, а в някои случаи, съответна обосновка за невъзможността да се предприемат подобни мерки в конкретния случай.

Handwritten signature

3.6. Инструменти за управление на риска

Предложената методика за управление на риска ще използва голям набор от инструменти и практики за постигане на целите на проекта. Всички инструменти са детайлно описани по-горе в частта методологии за управление на риска. Предлагаме съкратен списък на инструментите за управление на риска:

- Среци за планиране и анализ;
- Преглед на документация;
- Анализ на контролни списъци;
- Анализ на допусканията;
- Анализ на силни и слаби страни;
- Експертна оценка;
- Оценка на вероятността и влиянието на рисковете;
- Матрица на вероятност и влияние;
- Оценка на качеството на информацията за рисковете;
- Категоризация на рисковете;
- Оценка на неотложността на рисковете;
- Стратегии към негативни рискове и заплахи;
- Стратегии към позитивни рискове и благоприятни възможности;
- Стратегии за действие при непредвидени ситуации;
- Преоценка на рисковете;
- Одити на рисковете;
- Анализ на отклонения и тенденции;
- Анализ на резервите;
- Среци за обсъждане на състоянието.

За подпомагане поддържането на регистъра на рисковете се използва специализиран портал за управление на проекта, разработен от нас на базата на Microsoft Office SharePoint:

ID	Име	Сериозност	Вероятност	Влияние	Статус	Дата
1	Риск от това решение да не бъде прието от клиентите	Висока	Висока	Висока	Активен	15.12.2011
2	Риск от увеличаване на цените на материалите	Средна	Средна	Средна	Активен	15.12.2011
3	Риск от недостатъчно квалифициран персонал	Средна	Средна	Средна	Активен	15.12.2011
4	Риск от забавяне на доставките на оборудване	Средна	Средна	Средна	Активен	15.12.2011
5	Риск от промяна в изискванията на клиента	Средна	Средна	Средна	Активен	15.12.2011
6	Риск от промяна в изискванията на клиента	Средна	Средна	Средна	Активен	15.12.2011
7	Риск от промяна в изискванията на клиента	Средна	Средна	Средна	Активен	15.12.2011
8	Риск от промяна в изискванията на клиента	Средна	Средна	Средна	Активен	15.12.2011
9	Риск от промяна в изискванията на клиента	Средна	Средна	Средна	Активен	15.12.2011
10	Риск от промяна в изискванията на клиента	Средна	Средна	Средна	Активен	15.12.2011
11	Риск от промяна в изискванията на клиента	Средна	Средна	Средна	Активен	15.12.2011
12	Риск от промяна в изискванията на клиента	Средна	Средна	Средна	Активен	15.12.2011
13	Риск от промяна в изискванията на клиента	Средна	Средна	Средна	Активен	15.12.2011
14	Риск от промяна в изискванията на клиента	Средна	Средна	Средна	Активен	15.12.2011
15	Риск от промяна в изискванията на клиента	Средна	Средна	Средна	Активен	15.12.2011
16	Риск от промяна в изискванията на клиента	Средна	Средна	Средна	Активен	15.12.2011
17	Риск от промяна в изискванията на клиента	Средна	Средна	Средна	Активен	15.12.2011
18	Риск от промяна в изискванията на клиента	Средна	Средна	Средна	Активен	15.12.2011

4. Конкретни рискове

4.1. Анализ на конкретни рискове, идентифицирани от Възложителя

№	Описание	Последствия и сфера на влияние	Отговорник	Обща експозиция	Степен на значимост	Вероятност от настъпване	Въздействие	Мерки за минимизиране или предотвратяване на негативното въздействие и тяхната обосновка
1	Недобра комуникация между екипите на Възложителя и Изпълнителя по време на аналитичните етапи на проекта.	<p>В резултат на което може да се получи некачествено и забавено изпълнение на етапа за анализ и проектиране.</p> <p>Забавяне изпълнението на задачите по проекта.</p> <p>Проблеми с дефинирането на правилните изисквания при реализацията.</p> <p>Бавно вземане на решения.</p> <p>Некачествените и непълни изисквания могат до доведат до влагане на време и ресурс за разработка в погрешна посока.</p> <p>Сферата на влияние е в рамките на всички етапи на проекта, като</p>	Изпълнителя Възложителя	10	Висока	Ниска (21 - 40%)	<p>Забавяне на отговори, срещи и дейности критични за изпълнението на проекта.</p> <p>Непълни, неясни или често променящи се изисквания.</p> <p>Преразглеждане на взети решения.</p>	<p>Стриктно прилагане на предложената методология за изпълнение, която включва план за комуникация. Допълнително всяка седмица ще се провежда среща (разговор) между ключови представители на екипа.</p> <p>Регулярно предаване на доклади за напредъка по проекта. Определяне на хора за ключови роли в проектните екипи от двете страни.</p> <p>Прилагане на процедура за преглед (ревью) на резултатите от аналитичните етапи на проекта с участие на екипа на изпълнителя. Това ще спомогне еднозначното и пълно разбиране на изискванията и системния проект.</p> <p>По този начин възможността за пропуски в изпълнението се свежда до минимум и всяко</p>

№	Описание	Последствия и сфера на влияние	Отговорник	Обща експозиция	Степен на значимост	Вероятност от настъпване	Въздействие	Мерки за минимизиране или предотвратяване на негативното въздействие и тяхната обосновка
		може да се разпростре е върху бъдещи проекти за надграждане на системата и интегриране на други системи с нея.						отклонение ще бъде идентифицирано преди да е довело до реален проблем.
2	Ненавременно изпълнение на всяко от задълженията от страна на Изпълнителя	<p>Забавяне на изпълнението на поръчката;</p> <p>Необходимост за влагане на допълнителни ресурси за паралелна работа;</p> <p>Влошаване на качеството;</p> <p>Санкции.</p> <p>В зависимост от вида на забавянето, то може да повлияе конкретни задачи, етапи от изпълнението на поръчката или цялата поръчка. Значително забавяне на поръчката може да доведе до</p>	Изпълнителя	4	Съществена	Минимална (1 - 20%)	<p>Липса на съгласуван детайлен план-график;</p> <p>Липса на ангажимент от страна на Възложителя за зависимостите в графика;</p> <p>Забавяне на междинни срокове.</p>	<p>За минимизиране:</p> <p>Разработване и съгласуване на детайлен план-график за изпълнение;</p> <p>Ясно комуникиране на зависимостите с Възложителя;</p> <p>Определяне и проследяване на критичен път на задачите в проекта;</p> <p>Регулярен мониторинг и контрол на изпълнението;</p> <p>За минимизиране на последствията:</p> <p>Бързо предприемане на коригиращи действия при необходимост;</p> <p>Паралелна работа по различни задачи при необходимост да се компенсира забавяне;</p> <p>Влагане на допълнителни ресурси в някои от задачите</p>

№	Описание	Последствия и сфера на влияние	Отговорник	Обща експозиция	Степен на значимост	Вероятност от настъпване	Въздействие	Мерки за минимизиране или предотвратяване на негативното въздействие и тяхната обосновка
		забавяне на други зависещи от нея поръчки и проекти.						при необходимост да се компенсира забавяне. Предложените мерки съответстват на най-добрите и утвърдени световни практики за управление на проекти, като тези на PMI.
3	Неправилно и неефективно разпределяне ресурсите и отговорностите при изпълнението на договора	<p>Забавяне на изпълнението.</p> <p>Изпълнение с недостатъчно ниво на качество.</p> <p>Оскъпяване на изпълнението.</p> <p>Сферата на влияние може да бъде ограничена в рамките на конкретни задачи т.к. се предполага, че ще бъдат приложени мерки за предотвратяване и евентуално за минимизиране на последствията.</p>	Изпълнителя	4	Съществена	Минимална (1 - 20%)	Неясно дефинирани отговорности. Липса на план за наличност и капацитет на ресурсите.	<p>Ясно дефиниране на организационната структура, ролите и отговорностите – както за екипа на Изпълнителя, така и за Възложителя.</p> <p>Детайлно планиране на ресурсите по задачи в началото на проекта.</p> <p>Изпълнителят да даде препоръка за необходимите от страна на възложителя ресурси.</p> <p>Основният ресурс за изпълнение на поръчката са експертните екипи от страна на двете страни. Задачите по проекта трябва да бъдат обезпечени с необходимия ресурс, съобразно графика за изпълнение. Освен като наличност за конкретни</p>

№	Описание	Последствия и сфера на влияние	Отговорник	Обща експозиция	Степен на значимост	Вероятност от настъпване	Въздействие	Мерки за минимизиране или предотвратяване на негативното въздействие и тяхната обосновка
								задачи, експертите и екипите трябва да притежават необходимите знания, умения и подготовка за изпълнение на конкретните задачи.
4	Забавяне при изпълнение на проектните дейности, опасност от неспазване на срока за изпълнение на настоящата поръчка	Забавяне на следващи процедури, зависещи от настоящата; Необходимост за влагане на допълнителни ресурси за паралелна работа; Влошаване на качеството; Санкции. В зависимост от вида на забавянето, то може да повлияе конкретни задачи, етапи от изпълнението на поръчката или цялата поръчка. Значително забавяне на поръчката може да доведе до забавяне на други	Изпълнителя	4	Съществена	Минимална (1 - 20%)	Липса на съгласуван детайлен план-график; Липса на ангажимент от страна на Възложителя за зависимостите в графика; Забавяне на междинни срокове.	Мерки за предотвратяване: Разработване и съгласуване на детайлен план-график за изпълнение; Ясно комуникиране на зависимостите с Възложителя; Определяне и проследяване на критичен път на задачите в проекта; Регулярен мониторинг и контрол на изпълнението; Мерки за минимизиране на последствията: Бързо предприемане на коригиращи действия при необходимост; Паралелна работа по различни задачи при необходимост да се компенсира забавяне; Влагане на допълнителни ресурси в някои от задачите при необходимост да се компенсира забавяне.

000017

№	Описание	Последствия и сфера на влияние	Отговорник	Обща експозиция	Степен на значимост	Вероятност от настъпване	Въздействие	Мерки за минимизиране или предотвратяване на негативното въздействие и тяхната обосновка
		зависещи от нея поръчки и проекти.						Предложените мерки съответстват на най-добрите и утвърдени световни практики за управление на проекти, като тези на PMI.
5	Релевантност на предлаганите функционалности на системата	<p>Невъзможност на системата да функционира правилно;</p> <p>Невъзможност на системата да отговаря на изискванията за бързодействие;</p> <p>Затруднена интеграция с други системи.</p> <p>Благодарение на заложените процеси за осигуряване и контрол на качеството евентуални грешки би трябвало да бъдат установявани основно в етапа на разработка.</p> <p>Предвижда се установяване и коригиране на грешки и в етапа на тестване на системата.</p>	Изпълнител	4	Съществена	Минимална (1 - 20%)	Регистриране на инциденти в системата	<p>Изпълнителят ще следва стриктно процесите и добрите практики за разработка на софтуер. Прилагането на добри процеси и практики води до получаване на качествен продукт в предвидим срок и бюджет.</p> <p>При проектиране на системата се използват SOLID принципите за софтуерна архитектура – качеството ще бъде заложено в дизайна на системата.</p> <p>Процесите за софтуерна разработка включват мерки за осигуряване на качеството на процеса и контрол на качеството на разработвания продукт.</p> <p>Ще бъдат планирани и ще се провеждат всички необходими тестове. За целта ще бъдат разработени детайли тестови сценарии.</p>

№	Описание	Последствия и сфера на влияние	Отговорник	Обща експозиция	Степен на значимост	Вероятност от настъпване	Въздействие	Мерки за минимизиране или предотвратяване на негативното въздействие и тяхната обосновка
		Сферата на влияние следва да се ограничи в рамките на поръчката.						Предвидени са време и ресурс за отстраняване на грешки. Предвижда се прилагането на процедура и система за регистриране и отстраняване на грешки. Грешките ще бъдат отстраняване в предвидените срокове, съобразно техния приоритет.
6	Недостатъчна яснота по правната рамка и/или променяща се правна рамка по време на изпълнение на проекта	Може да доведе до концептуални непълноти и разминавания между цели и резултати. Промяна на изискванията, което ще доведе до увеличаване на сроковете или бюджета за реализиране на проекта. Промяна на вече изградени функционалности. Сферата на влияние е в рамките на поръчката.	Възложителя	5	Висока	Минимална (1 - 20%)	Възложителя променя изисквания по техническата спецификация след започване на проекта. Промени спрямо утвърдените в Етап 1 спецификации. Разширяване на обхвата. Отмяна на взети решения.	По време на фазата на анализ ще се анализират действащите норми и проектите за техните промени по време на реализацията. Ще се проведат проучвания за намеренията за промени свързани с проекта. Техническото изпълнение ще бъде съобразено с направеният анализ и всякакви промени ще бъдат съгласувани с Възложителя. Всяка заявка за промяна стриктно ще следва процедурата за управление на промените. Системата ще бъде гъвкава и в нея лесно биха могли да се реализират някои видове промени.

000019

№	Описание	Последствия и сфера на влияние	Отговорник	Обща експозиция	Степен на значимост	Вероятност от настъпване	Въздействие	Мерки за минимизиране или предотвратяване на негативното въздействие и тяхната обосновка
7	Неинформиране на Възложителя за всички потенциални проблеми, които биха могли да възникнат в хода на изпълнение на дейностите	Невъзможност за планиране и предприемане на адекватни и навременни коригиращи действия. Сферата на влияние е в рамките на конкретната поръчка.	Изпълнителя. Възложителя	4	Съществена	Минимална (1 - 20%)	Липса на план за комуникация или план за комуникация, в който не е указано как протича комуникацията за проблемите (issues); Липса на план за управление на проблемите; Неспазване на плановите за комуникация и управление на проблемите; Установяване на наличен проблем, който не е комуникиран.	Разработване и съгласуване на план за управление на проблемите; Поддържане и регулярно комуникиране на регистър на проблемите; Разработване и съгласуване на план за комуникация, който включва комуникация на проблемите; Включване на темата за проблемите в регулярните срещи и доклади по проекта; Публикуване на регистъра на проблемите в общодостъпна среда. Предложените мерки са съобразени със световно утвърдени добри практики за управление на проекти – конкретно PMBOK (Project Management Body of Knowledge) на PMI.
8.	Риск за администриране на системата след изтичане на периода на	Невъзможност на системата да функционира правилно;	Изпълнител Възложител	4	Съществена	Минимална (1 - 20%)	Липса на план за управление на проблемите; Неспазване на плановите за	Разработен и приет от Възложителя SLA след изтичане на гаранционната поддръжка.

№	Описание	Последствия и сфера на влияние	Отговорник	Обща експозиция	Степен на значимост	Вероятност от настъпване	Въздействие	Мерки за минимизиране или предотвратяване на негативното въздействие и тяхната обосновка
	гаранционна поддръжка	Невъзможност на потребителите да работят със системата					комуникация и управление на проблемите.	

4.2. Анализ на допълнителни, относими към поръчката рискове

№	Описание	Области на проявяване, ефект върху изпълнението и възможни последици	Отговорник	Обща експозиция	Степен на значимост	Вероятност от настъпване	Индикатор	Мерки за минимизиране или предотвратяване на негативното въздействие и тяхната обосновка
1	Неочаквани промени в изискванията поради външни причини – закони, стандарти, оперативни решения на ръководни органи на страната, международни задължения и др.	Промяна на изискванията, което ще доведе до увеличаване на сроковете или бюджета за реализиране на проекта. Промяна на вече изградени функционалности. Сферата на влияние е в рамките на поръчката.	Възложителя	5	Висока	Минимална (1 - 20%)	Възложителя променя изисквания по техническата спецификация след започване на проекта. Промени спрямо утвърдените в Етап 1 спецификации. Разширяване на обхвата. Отмяна на взети решения.	По време на етапа на анализ ще се направи анализ на действащите норми и проектите за техните промени по време на реализацията. Ще се проведат проучвания за намеренията за промени свързани с проекта. Техническото изпълнение ще бъде съобразено с направеният анализ и всякакви промени ще бъдат съгласувани с Възложителя. Всяка заявка за промяна стриктно ще следва процедурата за управление на промените. Системата е гъвкава и в нея лесно биха могли да се реализират някои видове промени.
2	Слаба ангажираност на представители на крайните потребители на системата.	Получаване на непълни и неясни изисквания. Забавяне при предоставянето на информация. Трудно внедряване на системата в реалната	Възложителя	6	Умерена	Ниска (21 - 40%)	Потребителите подават неясна, непълна или противоречива информация. Забавят се отговори и решения. Крайните потребители не използват	Определяне на конкретни ключови потребители за съответните области още в началото на проекта. По време на обученията ще се проведе максимално запознаване на потребителите с функционалностите на

№	Описание	Области на проявяване, ефект върху изпълнението и възможни последици	Отговорник	Обща експозиция	Степен на значимост	Вероятност от настъпване	Индикатор	Мерки за минимизиране или предотвратяване на негативното въздействие и тяхната обосновка
		работа на организацията. Сферата на влияние е в рамките на поръчката, но може да се разпростре и при интегрирането с други системи, в случай, че останат незабелязани дефекти.					новоизградената система.	системата и ползата за тях от нея. В обучението трябва да бъдат включени точните потребители/служители, които ще продължат да имат отношение към експлоатацията на системата.
	Липса на задълбоченост при изследването и описанието на бизнес процесите и данните	Недостатъчно добри изисквания към системата; Пропуснати изисквания; Неадекватна архитектура и оразмеряване. В случай на неадекватна архитектура, сферата на влияние може да се разпростре и върху други системи, чрез поставяне на неподходящи	Изпълнителя Възложителя	10	Висока	Ниска (21 - 40%)	Непълна и некачествена спецификация на изисквания; Изисквания, които не са SMART (Specific, Measurable, Attainable, Realisable, Traceable); Участие в изпълнението на ключови експерти, които не са добре запознати с предметната област.	Участието в проекта на правилните експерти от двете страни е решаващо за постигане на задълбоченост при изследването и описанието на бизнес процесите и данните. Особено важно е експертите на Възложителя да имат предвидено време за задачи по проекта и тези задачи да са с висок приоритет. Прилагане на процедура за преглед (ревью) на резултатите от аналитичните етапи на проекта с участие на екипа на изпълнителя. Това ще спомогне

№	Описание	Области на проявяване, ефект върху изпълнението и възможни последици	Отговорник	Обща експозиция	Степен на значимост	Вероятност от настъпване	Индикатор	Мерки за минимизиране или предотвратяване на негативното въздействие и тяхната обосновка
		изисквания за интеграция.						еднозначното и пълно разбиране на изискванията. За верифициране на изискванията, бизнес процесите и данните може да си използва съпоставяне със сходни проекти в други държави. Освен, че ще спомогне за гарантиране на качество, използването на добрия опит също може да улесни и ускори изпълнението.
3	Невъзможност за провеждане или сериозни затруднения за провеждане на ефективни тестове.	Липса на сигурност в безпроблемното функциониране на системата. Сферата на влияние се ограничава в Дейност 2 по настоящата поръчка.	Изпълнителя Възложителя	3	Умерена	Минимална (1 - 20%)	Документирана невозможност за провеждане на тестове. Голям брой грешки във внедрената система по-висок от допустимия праг.	По време на цялата реализация Изпълнителят ще провежда вътрешни хардуерни и софтуерни тестове на всички елементи преди тяхното предаване за тестване от Възложителя. Всички съвместни тестове ще бъдат предварително проверени и системите ще са претърпели предварителни тестове преди заявяване на готовност. С прилагането на най-добрите практики за системна интеграция възможността за невозможност за провеждане

№	Описание	Области на проявяване, ефект върху изпълнението и възможни последици	Отговорник	Обща експозиция	Степен на значимост	Вероятност от настъпване	Индикатор	Мерки за минимизиране или предотвратяване на негативното въздействие и тяхната обосновка
								на ефективни тестове е сведена до минимум. Изпълнителят ще осигури достъп на Възложителя до неговата тестова среда. Осигуряване на необходимото време за провеждане на тестовите.
4	Затруднения при осигуряване в периода на обучение на подходящите за обучение лица.	Част от служителите не са преминали обучение и им липсват специфични умения за работа със системата. Да бъдат обучени служители, които няма да имат пряко отношение към експлоатацията на системата за сметка на такива, които имат пряко отношение. Сферата на влияние се свежда до настоящата поръчка. Рискът би се отразил на Дейност в и последващата	Възложителя	1	Ниска	Минимална (1 - 20%)	По-малък брой присъстващи на обученията от предвиденото. Включените в обучение служители нямат определена роля по отношение на експлоатацията на системата.	Преди определяне на обучаемите, Възложителят трябва да определи служителите, които ще имат отношение към експлоатацията и поддръжката на системата и техните роли и отговорности. Изпълнителят ще предостави препоръки към обучаемите за да може Възложителят да подбере най-добрия възможен екип, които да бъде обучен. Графикът за обучения ще бъде направен няколко седмици преди провеждането на обученията. По време на изготвяне на графика всеки служител ще потвърди възможността си за участие. Обучението може да бъде

№	Описание	Области на проявяване, ефект върху изпълнението и възможни последици	Отговорник	Обща експозиция	Степен на значимост	Вероятност от настъпване	Индикатор	Мерки за минимизиране или предотвратяване на негативното въздействие и тяхната обосновка
		експлоатация на системата.						заснето и публикувано в портала.
5	Поява не изяснени изисквания в късен етап от проекта и промяна на вече утвърдени изисквания.	Влагане на допълнителни ресурси. Забавяне. Предизвикване на промени в други зависими системи. Сферата на влияние се разпростира върху етапите на разработка и тестване от настоящата поръчка	Изпълнителя Възложителя	12	Съществена	Средна (41 - 60%)	Промяна на утвърдени изисквания. Възникване на нови изисквания след фазата на дизайн.	Добро документиране на бизнес анализа и дизайна на системите. Прототипиране. Доброто документиране е предпоставка за еднозначно и пълно разбиране на изискванията. Комуникация при всяка една итерация за разработка. Добрата комуникация предполага, че неяснотите ще бъдат изяснявани на момента. Одобрение на детайлната техническа спецификация (Системния проект) от Възложителя. Допускане на промени само за най-важните изисквания – за да не се налагат значителни преработки и забавяния в следствие на тях.
6	Да се изгради система, която не отговаря на бизнес процесите и изискванията на възложителя.	Невъзможност за реална експлоатация на системата и постигане на целите на проекта. Ще окаже	Изпълнителя Възложителя	5	Висока	Минимална (1 - 20%)	Нееднозначно разбиране на изисквания. Поява на липсващи изисквания.	Задълбочаване на разработката едва след съгласуване на изискванията с Възложителя. Това ще предотврати влагането на усилия в разработка по неправилни изисквания.

№	Описание	Области на проявяване, ефект върху изпълнението и възможни последици	Отговорник	Обща експозиция	Степен на значимост	Вероятност от настъпване	Индикатор	Мерки за минимизиране или предотвратяване на негативното въздействие и тяхната обосновка
		влияние върху целия проект.						Прилагане на итеративен подход за разработка с участие на представители на Възложителя в церемониите по приключване на спринтовете (демо). Регулярното участие на Възложителя предполага навременна обратна връзка. Одобрение на детайлната техническа спецификация (Системния проект) от Възложителя.
7	Възникване на проблеми с натоварването на системата - недостатъчен хардуер и съответно лицензи за да се покрие натоварването на системата.	Прекъсване в работата или неприемливо бавна работа на системата. Влияние върху изпълнението на Дейност 2 и бъдещата експлоатация на системата.	Възложителя	4	Съществена	Минимална (1 - 20%)	Голямо натоварване, водещо до прекъсване в работата или неприемливо бавна работа на системите.	Разработване и готовност за прилагане на план за скалиране на хардуерната и софтуерна инфраструктура. Базиране на оценките на инфраструктурния модел върху прогнози за натоварване.
8	Информацията в системата да не бъде достатъчно защитена.	Изтичане на информация. Спиране на работа на системата. Влиянието се разпростира върху	Изпълнителя Възложителя	4	Съществена	Минимална (1 - 20%)	Липса на стратегия и процедури за защита на информацията. Неприлагане на процедурите за защита на информацията.	Разработване на стратегия за защита на информацията. Прилагане на стратегията за защита при инсталацията на експлоатационната среда. Редовно изпълнение на процедурите за защита.

№	Описание	Области на проявяване, ефект върху изпълнението и възможни последици	Отговорник	Обща експозиция	Степен на значимост	Вероятност от настъпване	Индикатор	Мерки за минимизиране или предотвратяване на негативното въздействие и тяхната обосновка
		системата, предмет на поръчката.						
9	Забавяне при вземането на решения или промяна на взети решения.	Закъснения. Необходимост от преправяне на вече извършена работа. Сферата на влияние е в рамките на поръчката.	Възложителя	3	Умерена	Минимална (1 - 20%)	Забавяне на отговори. Забавяне на решения. Промяна на взети решения.	Изготвяне и спазване на план за комуникация. Изготвяне и спазване на процедура за управление на промените. (И двете са част от плана за управление на проекта.)
10	Невъзможност за администриране и конфигуриране на техническа и софтуерна инфраструктура.	Невъзможност за извършване на инсталации и промяна на конфигурации. Необходимост от извършване на преинсталации. Загуба на информация. Влияние в рамките на поръчката.	Възложителя	3	Умерена	Минимална (1 - 20%)	Невъзможен администраторски достъп до инфраструктурни системи	Разработване и прилагане на процедури за администриране и конфигуриране. Осигуряване на отдалечен достъп до ИТ инфраструктурата.
11	Поставяне на неизпълними, абстрактни или неясно дефинирани изисквания.	Невъзможност за изпълнение или погрешно изпълнение. Забавяне на	Възложителя	4	Съществена	Минимална (1 - 20%)	Нееднозначно разбиране на изисквания.	Въвличане на служители от съответната бизнес дирекция при детайлизиране на системата. Разработване на ясни,

№	Описание	Области на проявяване, ефект върху изпълнението и възможни последици	Отговорник	Обща експозиция	Степен на значимост	Вероятност от настъпване	Индикатор	Мерки за минимизиране или предотвратяване на негативното въздействие и тяхната обосновка
		детайлизирането и изграждане на лош дизайн. Влияние в рамките на поръчката. Допускането на лош дизайн може да се отрази върху бъдещите интеграции със системата и възможността за нейното надграждане.						еднозначни и достатъчно подробни спецификации. Реализиране на системата по гъвкав начин, чрез който лесно да могат да се реализират евентуални промени. Стартиране на разработка след наличие на достатъчно детайлни изисквания.
12	Неустановена практика по процедурата за промени	Забавяне на искани промени поради бавна или неработеща процедура, а и от там на сроковете по проекта. Влияние в рамките на поръчката.	Изпълнителя Възложителя	12	Умерена	Голяма (61 - 80%)	Липса на двустранно утвърдена процедура за управление на промени.	Въвеждане на ясна процедура за управление на промените. Приоритизиране на исканията за промяна и реализиране само на най-приоритетните при спазване на обхвата на проекта.
13	Екипът на Възложителя е недостатъчен, не е на разположение за проекта или задачите по проекта са с нисък приоритет	Забавяне вземането на решения и като следствие забавяне на работата по проекта. Вземане на грешни решения и като	Възложителя	4	Съществена	Минимална (1 - 20%)	Забавяне на отговори. Забавяне вземането на решения. Промяна на вече взети решения.	Регламентиране по вътрешен ред на приоритетите при изпълнение на работата на екипа. Предоставяне на допълнителни членове към екипа с необходимите компетенции.

№	Описание	Области на проявяване, ефект върху изпълнението и възможни последици	Отговорник	Обща експозиция	Степен на значимост	Вероятност от настъпване	Индикатор	Мерки за минимизиране или предотвратяване на негативното въздействие и тяхната обосновка
	спрямо останалата работа.	резултат необходимост от повторно извършване на части от работата по проекта. Влияние в рамките на поръчката.						Осигуряване на допълнителна ангажираност на експертния персонал на Възложителя. Привличане на външни консултанти.
14	Трудности при осигуряване на условия за извършване на приемни тестове	Забавяне на приемните тестове. Влияние в рамките на поръчката.	Възложителя	3	Умерена	Минимална (1 - 20%)	Липса на потвърждение за готовност за приемни тестове.	Предварително изготвени изисквания за провеждане на приемни тестове и осигуряване на условия. Осигуряване на резерв от време за извършването на приемните тестове.
15	Системата да не бъде подготвена за посрещане на заплахи	Срив на системата в случай на проявяване на заплаха.	Изпълнителя Възложителя	3	Умерена	Минимална (1 - 20%)	Липса на стратегия за резервираност. Липса на инфраструктура за резервираност.	Разработване на стратегия за резервираност, която включва: анализ на рисковете за изгражданото решение, възможните източници на заплаха, възможните слаби места в структурата на системата и т.н. Осигуряване на необходимата инфраструктура за резервираност. Наличие на процедури за възстановяване на системата.

№	Описание	Области на проявяване, ефект върху изпълнението и възможни последици	Отговорник	Обща експозиция	Степен на значимост	Вероятност от настъпване	Индикатор	Мерки за минимизиране или предотвратяване на негативното въздействие и тяхната обосновка
16	Опасност от надхвърляне обхвата на изискванията за софтуера	Забавяне на бюджета Надвишаване на въздействието на качеството на изпълнение на проекта. Влияние върху конкретната поръчка.	Изпълнител	8	Съществена	Ниска (21 - 40%)	Доработване на нови функционалности, които са извън утвърдената спецификация; Голям брой заявки за промяна	Разбрани, съгласувани и утвърдени изисквания в Етап 1 на проекта; Внимателно комуникиране на изискванията; Представяне на прототип; Въвличане на ключови потребители по итерации и разработка. Одобрение на детайлната техническа спецификация (Системния проект) от Възложителя.
17	Невъзможност за пълно тестване на системата	Наличие на дефекти в системата. Влияние върху конкретната поръчка.	Изпълнител	10	Висока	Ниска (21 - 40%)	Регистриране на инциденти в системата; Висок брой на дефектите	Прецизно планиране на тестването; Разработване на пълен набор от тестови сценарии; Използване на инструменти за автоматизирано тестване; Осигуряване на пълна тестова среда.
18	Невъзможност за изпълнение на параметрите на SLA.	Превишаване на времето за престой на системата. Сферата на влияние е както за конкретната система, така за системите, обслужващи регистрите, които	Изпълнител, Възложител	10	Висока	Ниска (21 - 40%)	Неспазване на сроковете за реакция и острояване на проблеми; Надвишаване на времената за престой и броя на планираните спирия на системата.	Сключване и спазване на SLA за работата на техническата инфраструктура т.к. без работеща инфраструктура няма как да работи самата система; Еднозначно регламентиране на ангажиментите по SLA за Изпълнителя и Възложителя;

000231

№	Описание	Области на проявяване, ефект върху изпълнението и възможни последици	Отговорник	Обща експозиция	Степен на значимост	Вероятност от настъпване	Индикатор	Мерки за минимизиране или предотвратяване на негативното въздействие и тяхната обосновка
		подават информация както и румънския компонент с който се обменя информация.						Стриктно спазване на процедурите поддръжка; Прилагане на мероприятия за проактивна поддръжка.
19	Липсата на подходящо електрозахранване	Невъзможност за пускане в експлоатация на оборудването	Възложителя	5	Висока	Минимална (1 - 20%)	Възложителя не е предвидил необходимото електроразпределително табло снабдено с подходящи прекъсвачи за свързване на оборудването	След подписване на договор за изпълнение ще бъдат предоставени необходимите изисквания за да може Възложителят да осигури необходимото електрозахранване
20	Липса на достатъчно на брой свободни портове в мрежовото оборудване на Възложителя за свързване на доставеното оборудване	Невъзможност за пускане в експлоатация на оборудването	Възложителя	3	Умерена	Минимална (1 - 20%)	Възложителя не е предвидил необходимите портове за свързване на оборудването към текущата му среда	След подписване на договор за изпълнение ще бъдат предоставени необходимите изисквания за да може Възложителят да осигури необходимото за мрежовата свързаност на оборудването

№	Описание	Области на проявяване, ефект върху изпълнението и възможни последици	Отговорник	Обща експозиция	Степен на значимост	Вероятност от настъпване	Индикатор	Мерки за минимизиране или предотвратяване на негативното въздействие и тяхната обосновка
21	Липсата на достатъчно широка и висока врата на сървърното помещение	Невъзможност за пускане в експлоатация на оборудването поради невъзможността да се внесе в сървърното помещение	Възложител	3	Умерена	Минимална (1 - 20%)	Възложителя не е предвидил необходимите размери на вратите в сървърното помещение	След подписване на договор за изпълнение ще бъдат предоставени размерите на оборудването, за да може Възложителят да осигури в случай на нужда необходимия отвор за да може да се внесе оборудването в сървърното помещение
22	Липсата на достатъчно подсилен под в сървърното помещение	Възможност да пропадне шкафа или да се наклони, което може да доведе до повреда на техниката	Възложител	3	Умерена	Минимална (1 - 20%)	Възложителя не е предвидил достатъчно масивен под, който може да издържи теглото на оборудването	След подписване на договор за изпълнение ще бъдат предоставено теглото на оборудването, за да може Възложителят да укрепи в случай на необходимост пода в сървърното помещение

000.235



към

ТЕХНИЧЕСКО ПРЕДЛОЖЕНИЕ

за участие в открита процедура за възлагане
на обществена поръчка с предмет:

**„Доставка на интегрирана система за корабна инспекция –
софтуерен пакет с електронна база данни и оборудване“**



Подробен график



(в който са конкретизирани сроковете за изпълнение
на всяка дейност от настоящата поръчка)



2018 г.

77

Съдържание

1. Въведение.....	3
2. Срок за изпълнение на поръчката.....	3
3. Етапи, срокове и резултати	3
3.1. Анализ и проектиране.....	3
3.2. Конфигуриране и адаптиране.....	4
3.3. Доставка и внедряване.....	5
4. Подробен план-график, в който са конкретизирани сроковете за изпълнение на всяка дейност от настоящата поръчка.....	6
5. Логическа връзка и зависимости между задачите	7





1. Въведение

Настоящото приложение представя визията на Кандидата за етапите на изпълнение на поръчката и сроковете за тях.

Началото на проекта е условно, приложен е примерен план-график, с начална дата 30.07.2018 г.

2. Срок за изпълнение на поръчката


Срокът за изпълнение на настоящата поръчка е 9 (девет) месеца от датата на подписване на договора между Възложителя и Изпълнителя.

№	Етап	Срок
1	Анализ и проектиране	2 месеца след подписване на договор - 28.09.2018 г
2	Конфигуриране и адаптиране	8 месеца след подписване на договор - 29.03.2019 г.
3	Доставка и внедряване	9 месеца след подписване на договор - 30.04.2019 г.



3. Етапи, срокове и резултати

Изпълнението на поръчката ще бъде извършено на следните етапи:


- Етап 1: Анализ и проектиране;
 - Етап 2: Конфигуриране и адаптиране;
 - Етап 3: Доставка и внедряване.
- 


Изпълнението на някои от етапите ще се застъпва.

3.1. Анализ и проектиране

В началото на изпълнението на етапа, Изпълнителят ще изготви детайлен график на изпълнение на поръчката, който ще бъде съгласуван с Възложителя.

Етапът обхваща изпълнението на следните задачи:

- Детайлно проучване на изискванията към новите функционалности на системата;
 - Дефиниране на детайлните изисквания и бизнес процеси, които трябва да се реализират в системата;
 - Дизайн на системата, хардуерната и комуникационната инфраструктура;
 - Проектиране на необходимите промени и доработки в модули и функционалности, както и тяхната конфигурация, въз основа на анализа и изискванията;
 - Проектиране и специфициране на интеграционен интерфейс с румънския компонент на ИИС и модел на обменните данни.
 - Изготвяне на план за техническа реализация;
 - Определяне на потребителския интерфейс;
 - Изготвяне на подробна Спецификация на софтуерните изисквания;
 - Изготвяне на Системна архитектура;
- 

- 
- Изготвяне на Спецификация на интеграционен интерфейс с румънския компонент на ИИС.

Детайлното проучване на системните и софтуерни изисквания ще обхваща всички компоненти в обхвата на поръчката, свързани с информационната система и нейното внедряване.


При проучването и анализа на изискванията и при проектирането на системата Изпълнителят ще вземе предвид хармонизираните процедури за координация и надзор на речния трафик, разработени по проекта.

При документирането на изискванията и системната архитектура, ще се използва структурирано описание и стандартен език за моделиране – UML нотация. Структурираните изисквания ще бъдат основа за създаване на тестови случаи за приемане на системата.

В резултат от изпълнението на етапа, ще бъдат предадени за одобрение от Възложителя следните документи:

- Спецификация на софтуерните изисквания
- Системната архитектура
- Спецификацията на интеграционен интерфейс с румънския компонент

Те ще подлежат на одобрение от Възложителя.



Всички документи разработени в Етап 1 ще бъдат утвърдени от Възложителя и ще бъдат основа за реализиране на следващите етапи. В срок от 5 работни дни Възложителят трябва да утвърди документите или да върне коментари по тях.

3.2. Конфигуриране и адаптиране

Въз основа на спецификациите на изискванията и на съгласуваните елементи от дизайна, описани по-горе, по време на етапа на конфигуриране и адаптиране, Кандидатът ще реализира или адаптира модулите и функционалностите на системата.

Етапът на конфигуриране и адаптиране включва изпълнението на следните задачи:

- Конфигуриране на модулите на системата съгласно изискванията на техническото задание и техническата спецификация;
- Актуализиране на спецификациите, разработени в рамките на Етап 1
- Подготвяне на тестови случаи за приемане на информационната система, съобразени с одобрената от Възложителя спецификации на изискванията, включващи:
 - Функционални тестове;
 - Интеграционни тестове.
- Провеждане на вътрешни тестове на системата (в тестова среда на разработчика);
- Създаване специализирани изгледи на данните за справочни цели, инсталиране и конфигуриране на справочната система;
- Конфигуриране на примерни справки.
- Разработване на документация за потребителите и администраторите на системата;
- Изготвяне на детайлни тестови сценарии за приемане на системата;

Handwritten signature

В резултат от изпълнението на етапа, ще бъдат изготвени и предадени за одобрение от Възложителя следните документи:

- Инструкции за инсталация и скриптове за създаване на базата данни, включващи:
 - Детайлно описание на базата данни;
 - Описание на софтуерните модули.
 - Изходен код на разработените в обхвата на поръчката модули и функционалности;
 - Документация на системата, вкл.:
 - Ръководство за потребителя;
 - Ръководство за администратора;
 - План за тестване;
 - Тестови сценарии за приемане на системата.
- Handwritten signature*

Всички материали разработени в Етап 2 следва да бъдат утвърдени от Възложителя и ще бъдат основа на системата, която ще бъде внедрена в експлоатация.

3.3. Доставка и внедряване

В този етап Изпълнителят ще осигури експлоатационната среда на информационната система като ще изпълни следните задачи:

- Handwritten signature*
- Инсталиране, конфигуриране и настройка на програмните компоненти на системата в условията на експлоатационната среда на Възложителя.
 - Въвеждане на всички номенклатури, класификатори и конфигурации (ако е необходимо въвеждането на такива)
 - Въвеждане на всички потребители, които няма да се определят с работен процес посредством системата.
 - Провеждане на приемателни тестове;
 - Доставка, инсталиране и конфигуриране на ИКТ инфраструктура за ИИС за инспекция на кораби:
 - Доставка на оборудването в локации на Възложителя
 - Инсталация на сървърен шкаф
 - Монтаж на UPS в сървърния шкаф
 - Монтаж на блейд шаси в сървърния шкаф
 - Монтаж на споделен сторидж в сървърния шкаф
 - Инсталация на операционни системи
 - Инсталиране и конфигуриране на 2 x Тауър сървър
 - Функционални тестове на хардуерните ресурси
 - Актуализиране на системната документация, предадена в Етап 2;
 - Разработване на план за обучения;
 - Организиране и провеждане на обучения за следните групи и ползватели на софтуерното решение:
 - Ключови потребители – до 30 човека, разпределени в две учебни групи;
 - Администратори на системата – до 5 човека.

В резултат от изпълнението на етапа, ще бъдат изготвени и предадени за одобрение от Възложителя следните документи:

- Актуализирана системна документация (в случай на необходимост);
- Handwritten signature*

- План за обучения;
- Учебни материали;
- Протоколи за проведени обучения;
- Протокол за успешно проведени приемни тестове;
- Протоколи за доставка, инсталация и конфигурация на ИКТ инфраструктурата по Дейност 2;
- Протоколи за доставка, инсталация и конфигурация на базов софтуер.

4. Подробен план-график, в който са конкретизирани сроковете за изпълнение на всяка дейност от настоящата поръчка

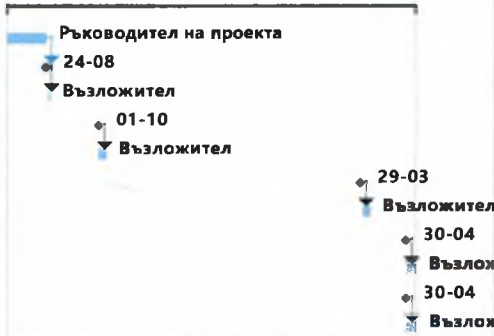
ID	Task Name	Duration	Start	Finish	Predec	Qtr 3, 2018			Qtr 4, 2018			Qtr 1, 2019			Qtr 2, 2019		
						Jul	Aug	Sep	Oct	Nov	Dec	Jan	Feb	Mar	Apr	May	Jun
17	Спецификацията на интеграционен интерфейс с румънския компонент	0 days	Fri 28-09-18	Fri 28-09-18	14				28-09								
18	Протокол за приключен Етап 1	0 days	Mon 01-10-18	Mon 01-10-18					01-10								
19	Възложителят утвърждава документите	5 days	Mon 01-10-18	Fri 05-10-18	15,16,1				Възложител								
20	Утвърдени документи	0 days	Fri 05-10-18	Fri 05-10-18	19				05-10								
21	Етап 2: Конфигуриране и адаптиране	130 days	Mon 01-10-18	Fri 29-03-19	2												
22	Конфигуриране на модулите на системата съгласно изискванията на техническото задание и техническата спецификация;	120 days	Mon 01-10-18	Fri 15-03-19													
23	Създаване специализирани изгледи на данните за справочни цели, инсталиране и конфигуриране на справочната система;	6 mons	Mon 01-10-18	Fri 15-03-19													Бизнес анализа
24	Конфигуриране на примерни справки.	6 mons	Mon 01-10-18	Fri 15-03-19													Бизнес анализа
25	Актуализиране на спецификациите, разработени в рамките на Етап 1	5 days	Mon 11-03-19	Fri 15-03-19	22FF												Бизнес анализа
26	Подготвяне на тестови случаи за приемане на информационната система, съобразени с одобрената от Възложителя спецификации на	5 days	Mon 11-03-19	Fri 15-03-19	25SS												
27	Функционални тестове;	5 days	Mon 11-03-19	Fri 15-03-19													Експерт по Кон
28	Интеграционни тестове.	5 days	Mon 11-03-19	Fri 15-03-19													Експерт по Кон
29	Провеждане на вътрешни тестове на системата (в тестова среда на разработчика);	3 mons	Mon 24-12-18	Fri 15-03-19	22FF												Експерт по Кон
30	Разработване на документация за потребителите и администраторите на системата;	10 days	Mon 18-03-19	Fri 29-03-19	29												Специалист п
31	Изготвяне на детайлни тестови сценарии за приемане на системата;	10 days	Mon 18-03-19	Fri 29-03-19	26												Експерт по Кс

ID	Task Name	Duration	Start	Finish	Predec	Qtr 3, 2018			Qtr 4, 2018			Qtr 1, 2019			Qtr 2, 2019			
						Jul	Aug	Sep	Oct	Nov	Dec	Jan	Feb	Mar	Apr	May	Jun	
32	Исходен код на разработените в обхвата на поръчката модули и функционалности;	0 days	Fri 29-03-19	Fri 29-03-19	31													29-03
33	Документация на системата, вкл.:	0 days	Fri 29-03-19	Fri 29-03-19	30													29-03
34	Ръководство за потребителя;	0 days	Fri 29-03-19	Fri 29-03-19														29-03
35	Ръководство за администратора;	0 days	Fri 29-03-19	Fri 29-03-19														29-03
36	Инструкции за инсталация и скриптове за създаване на базата данни;	0 days	Fri 29-03-19	Fri 29-03-19	22													29-03
37	Детайлно описание на базата данни;	0 days	Fri 29-03-19	Fri 29-03-19														29-03
38	Описание на софтуерните модули;	0 days	Fri 29-03-19	Fri 29-03-19														29-03
39	План за тестване;	0 days	Fri 29-03-19	Fri 29-03-19	31													29-03
40	Тестови сценарии за приемане на системата.	0 days	Fri 29-03-19	Fri 29-03-19	31													29-03
41	Протокол за приключване на Етап 2	0 days	Fri 29-03-19	Fri 29-03-19	36,32,3													29-03
42	Възложителят утвърждава документите	5 days	Mon 01-04-19	Fri 05-04-19	41													Възложител
43	Утвърдени документи	0 days	Fri 05-04-19	Fri 05-04-19	42													05-04
44	Етап 3: Доставка и внедряване	21.5 days	Mon 01-04-19	Tue 30-04-19	21													
45	Доставка, инсталиране и конфигуриране на ИКТ инфраструктура за ИИС за инспекция на кораби	15.5 days	Mon 01-04-19	Mon 22-04-19														
46	Доставка на оборудването в локация на Възложителя	1 day	Mon 01-04-19	Mon 01-04-19														Ръководител
47	Инсталация на сървърен шкаф	0.5 days	Tue 02-04-19	Tue 02-04-19	46													Специалист п
48	Монтаж на UPS в сървърния шкаф	0.5 days	Tue 02-04-19	Tue 02-04-19	47													Специалист п
49	Монтаж на блейд хаши в сървърния шкаф	1.5 days	Wed 03-04-19	Thu 04-04-19	48													Специалист
50	Монтаж на споделен сторидж в сървърния шкаф	1.5 days	Mon 01-04-19	Tue 02-04-19														Специалист п

Handwritten signature

ID	Task Name	Duration	Start	Finish	Precedence	Qtr 3, 2018			Qtr 4, 2018			Qtr 1, 2019			Qtr 2, 2019			
						Jul	Aug	Sep	Oct	Nov	Dec	Jan	Feb	Mar	Apr	May	Jun	
51	Инсталация на операционни системи	6 days	Thu 04-04-19	Fri 12-04-19	50,49													Специалист
52	Инсталиране и конфигуриране на 2 x Тауър сървъра	1 day	Fri 12-04-19	Mon 15-04-19	51													Специалист
53	Функционални тестове на хардуерните ресурси	5 days	Mon 15-04-19	Mon 22-04-19	52													Специалист
54	Инсталиране, конфигуриране и настройка на програмните компоненти на системата в условията на експлоатационната среда на	5 days	Mon 15-04-19	Mon 22-04-19	52													Специалист
55	Въвеждане на всички номенклатури, класификатори и конфигурации (ако е необходимо)	3 days	Mon 22-04-19	Thu 25-04-19	45													Специалист
56	Въвеждане на всички потребители, които няма да се определят с работен процес посредством	3 days	Mon 22-04-19	Thu 25-04-19	45													Специалист
57	Актуализиране на системната документация, предадена в Етап 2;	3 days	Mon 22-04-19	Thu 25-04-19	45													Бизнес аналитик
58	Разработване на план за обучения;	10 days	Mon 01-04-19	Fri 12-04-19														Специалист
59	Провеждане на приемни тестове	1 day	Thu 25-04-19	Fri 26-04-19	56													Експерт
60	Организиране и провеждане на обучения за следните групи и ползватели на софтуерното	3 days	Thu 25-04-19	Tue 30-04-19	56,58													Специалист
61	Ключови потребители ГРУПА 1	1 day	Thu 25-04-19	Fri 26-04-19														Специалист
62	Ключови потребители ГРУПА 2	1 day	Fri 26-04-19	Mon 29-04-19	61													Специалист
63	Администратори на системата	1 day	Mon 29-04-19	Tue 30-04-19	62													Системна
64	Актуализирана системна документация (в случай на необходимост);	0 days	Thu 25-04-19	Thu 25-04-19	57													25-04
65	План за обучения;	0 days	Fri 12-04-19	Fri 12-04-19	58													12-04
66	Учебни материали;	0 days	Fri 12-04-19	Fri 12-04-19	58													12-04
67	Протоколи за проведени обучения;	0 days	Tue 30-04-19	Tue 30-04-19	60													30-04

ID	Task Name	Duration	Start	Finish	Precedence	Qtr 3, 2018			Qtr 4, 2018			Qtr 1, 2019			Qtr 2, 2019			
						Jul	Aug	Sep	Oct	Nov	Dec	Jan	Feb	Mar	Apr	May	Jun	
68	Протокол за успешно проведени приемни тестове;	0 days	Fri 26-04-19	Fri 26-04-19	59													26-04
69	Протоколи за доставка, инсталация и конфигурация на ИКТ инфраструктурата по Дейност	0 days	Fri 26-04-19	Fri 26-04-19	59													26-04
70	Протоколи за доставка, инсталация и конфигурация на базов софтуер.	0 days	Fri 26-04-19	Fri 26-04-19	59													26-04
71	Протокол за приключване на Етап 3	0 days	Tue 30-04-19	Tue 30-04-19	64,65,66													30-04
72	Окончателен приемо предавателен протокол	0 days	Tue 30-04-19	Tue 30-04-19	71													30-04
73	Възложителят утвърждава документите	5 days	Tue 30-04-19	Tue 07-05-19	72													Възложител
74	Утвърдени документи	0 days	Tue 07-05-19	Tue 07-05-19	73													07-05
75	Докладване	201.5 days	Mon 30-07-19	Tue 07-05-19														
76	Подготовка на Въспителен доклад	1 mon	Mon 30-07-19	Fri 24-08-19														
77	Въспителен доклад	0 days	Fri 24-08-19	Fri 24-08-19	76													
78	Одобрение от Възложителя	1 day	Mon 27-08-19	Mon 27-08-19	77													
79	Междинен доклад Етап 1	0 days	Mon 01-10-19	Mon 01-10-19	72													
80	Одобрение от Възложителя	5 days	Mon 01-10-19	Fri 05-10-19	79													
81	Междинен доклад Етап 2	0 days	Fri 29-03-19	Fri 29-03-19	21													
82	Одобрение от Възложителя	5 days	Mon 01-04-19	Fri 05-04-19	81													
83	Междинен доклад Етап 3	0 days	Tue 30-04-19	Tue 30-04-19	44													
84	Одобрение от Възложителя	5 days	Tue 30-04-19	Tue 07-05-19	83													
85	Окончателен доклад	0 days	Tue 30-04-19	Tue 30-04-19	44													
86	Одобрение от Възложителя	5 days	Tue 30-04-19	Tue 07-05-19	85													



5. Логическа връзка и зависимости между задачите

Логическата връзка и зависимостите на задачите са включени и представени в примерния план-график, приложен към настоящия документ.

Handwritten signature

Handwritten signature

ID	Task Name	Duration	Start	Finish	Predecessors	2019				2020				2021					
						Q2	Q3	Q4	Q1	Q2	Q3	Q4	Q1	Q2	Q3	Q4	Q1	Q2	Q3
1	Подписване на договор	0 days	Mon 30-07-18	Mon 30-07-18															
2	Етап 1: Анализ и проектиране;	45 days	Mon 30-07-18	Mon 01-10-18															
3	Детайлен график на изпълнение на поръчката	0 days	Mon 30-07-18	Mon 30-07-18															
4	Съгласуване на детайлния график с Възложителя	1 day	Mon 30-07-18	Mon 30-07-18															
5	Детайлно проучване на изискванията към новите функционалности на системата;	10 days	Mon 30-07-18	Fri 10-08-18															
6	Дефиниране на детайлните изисквания и бизнес процеси, които трябва да се реализират в системата;	10 days	Mon 13-08-18	Fri 24-08-18	5														
7	Дизайн на системата, хардуерната и комуникационната инфраструктура;	17 days	Mon 20-08-18	Tue 11-09-18	6F5-5 days														
8	Проектиране на необходимите промени и доработки в модули и функционалности, както и тяхната конфигурация, въз основа на анализа и изискванията;	10 days	Wed 12-09-18	Tue 25-09-18	7														
9	Проектиране и специфициране на интеграционен интерфейс с румънския компонент на ИИС и модел на	10 days	Wed 12-09-18	Tue 25-09-18	7														
10	Изготвяне на план за техническа реализация;	3 days	Wed 26-09-18	Fri 28-09-18	9														
11	Определяне на потребителския интерфейс;	3 days	Wed 26-09-18	Fri 28-09-18	7,9														
12	Изготвяне на подробна Спецификация на софтуерните изисквания;	3 days	Wed 26-09-18	Fri 28-09-18	6,7,9														
13	Изготвяне на Системна архитектура;	3 days	Wed 26-09-18	Fri 28-09-18	7,9														
14	Изготвяне на Спецификация на интеграционен интерфейс с румънския компонент на ИИС.	3 days	Wed 26-09-18	Fri 28-09-18	7,9														
15	Спецификация на софтуерните изисквания	0 days	Fri 28-09-18	Fri 28-09-18	10,11,12														
16	Системната архитектура	0 days	Fri 28-09-18	Fri 28-09-18	13														
17	Спецификацията на интеграционен интерфейс с румънския компонент	0 days	Fri 28-09-18	Fri 28-09-18	14														
18	Протокол за приключен Етап 1	0 days	Mon 01-10-18	Mon 01-10-18															
19	Възложителят утвърждава документите	5 days	Mon 01-10-18	Fri 05-10-18	15,16,17														

Project: Danriss Schedule v1
Date: Mon 25-06-18

Task		Inactive Summary		External Tasks	
Split		Manual Task		External Milestone	
Milestone		Duration-only		Deadline	
Summary		Manual Summary Rollup		Progress	
Project Summary		Manual Summary		Manual Progress	
Inactive Task		Start-only			
Inactive Milestone		Finish-only			

000243

ID	Task Name	Duration	Start	Finish	Predecessor	2019				2020				2021					
						Q2	Q3	Q4	Q1	Q2	Q3	Q4	Q1	Q2	Q3	Q4	Q1	Q2	Q3
20	Утвърдени документи	0 days	Fri 05-10-18	Fri 05-10-18	19				05-10										
21	Етап 2: Конфигуриране и адаптиране	130 days	Mon 01-10-18	Fri 29-03-19	2														
22	Конфигуриране на модулите на системата съгласно изискванията на техническото задание и техническата спецификация;	120 days	Mon 01-10-18	Fri 15-03-19															
23	Създаване специализирани изгледи на данните за справочни цели, инсталиране и конфигуриране на справочната система;	6 mons	Mon 01-10-18	Fri 15-03-19															
24	Конфигуриране на примерни справки.	6 mons	Mon 01-10-18	Fri 15-03-19															
25	Актуализиране на спецификациите, разработени в рамките на Етап 1	5 days	Mon 11-03-19	Fri 15-03-19	22FF														
26	Подготвяне на тестови случаи за приемане на информационната система, съобразени с одобрената от Възложителя спецификации на изискванията,	5 days	Mon 11-03-19	Fri 15-03-19	25SS														
27	Функционални тестове;	5 days	Mon 11-03-19	Fri 15-03-19															
28	Интеграционни тестове.	5 days	Mon 11-03-19	Fri 15-03-19															
29	Провеждане на вътрешни тестове на системата (в тестова среда на разработчика);	3 mons	Mon 24-12-18	Fri 15-03-19	22FF														
30	Разработване на документация за потребителите и администраторите на системата;	10 days	Mon 18-03-19	Fri 29-03-19	29														
31	Изготвяне на детайлни тестови сценарии за приемане на системата;	10 days	Mon 18-03-19	Fri 29-03-19	26														
32	Изходен код на разработените в обхвата на поръчката модули и функционалности;	0 days	Fri 29-03-19	Fri 29-03-19	31														
33	Документация на системата, вкл.:	0 days	Fri 29-03-19	Fri 29-03-19	30														
34	Ръководство за потребителя;	0 days	Fri 29-03-19	Fri 29-03-19															
35	Ръководство за администратора;	0 days	Fri 29-03-19	Fri 29-03-19															
36	Инструкции за инсталация и скриптове за създаване на базата данни;	0 days	Fri 29-03-19	Fri 29-03-19	22														

Project: Danriss Schedule v1
Date: Mon 25-06-18

Task		Inactive Summary		External Tasks	
Split		Manual Task		External Milestone	
Milestone		Duration-only		Deadline	
Summary		Manual Summary Rollup		Progress	
Project Summary		Manual Summary		Manual Progress	
Inactive Task		Start-only			
Inactive Milestone		Finish-only			

Handwritten signature

Handwritten signature

Handwritten signature

Handwritten signature

000244

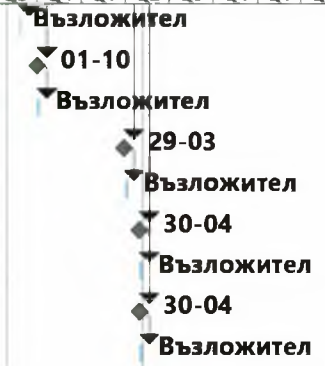
ID	Task Name	Duration	Start	Finish	Predecessor	2019				2020				2021						
						Q2	Q3	Q4	Q1	Q2	Q3	Q4	Q1	Q2	Q3	Q4	Q1	Q2	Q3	Q4
37	Детайлно описание на базата данни;	0 days	Fri 29-03-19	Fri 29-03-19						◆										
38	Описание на софтуерните модули;	0 days	Fri 29-03-19	Fri 29-03-19						◆										
39	План за тестване;	0 days	Fri 29-03-19	Fri 29-03-19	31					◆										
40	Тестови сценарии за приемане на системата.	0 days	Fri 29-03-19	Fri 29-03-19	31					◆										
41	Протокол за приключване на Етап 2	0 days	Fri 29-03-19	Fri 29-03-19	36,32,33					◆										
42	Възложителят утвърждава документите	5 days	Mon 01-04-19	Fri 05-04-19	41					◆										
43	Утвърдени документи	0 days	Fri 05-04-19	Fri 05-04-19	42					◆										
44	Етап 3: Доставка и внедряване	21.5 days	Mon 01-04-19	Tue 30-04-19	21					■										
45	Доставка, инсталиране и конфигуриране на ИКТ инфраструктура за ИИС за инспекция на кораби	15.5 days	Mon 01-04-19	Mon 22-04-19						■										
46	Доставка на оборудването в локации на Възложителя	1 day	Mon 01-04-19	Mon 01-04-19						■										
47	Инсталация на сървърен шкаф	0.5 days	Tue 02-04-19	Tue 02-04-19	46					■										
48	Монтаж на UPS в сървърния шкаф	0.5 days	Tue 02-04-19	Tue 02-04-19	47					■										
49	Монтаж на блейд шаси в сървърния шкаф	1.5 days	Wed 03-04-19	Thu 04-04-19	48					■										
50	Монтаж на споделен сторидж в сървърния шкаф	1.5 days	Mon 01-04-19	Tue 02-04-19						■										
51	Инсталация на операционни системи	6 days	Thu 04-04-19	Fri 12-04-19	50,49					■										
52	Инсталиране и конфигуриране на 2 x Тауър сървър	1 day	Fri 12-04-19	Mon 15-04-19	51					■										
53	Функционални тестове на хардуерните ресурси	5 days	Mon 15-04-19	Mon 22-04-19	52					■										
54	Инсталиране, конфигуриране и настройка на програмните компоненти на системата в условията на експлоатационната среда на Възложителя.	5 days	Mon 15-04-19	Mon 22-04-19	52					■										
55	Въвеждане на всички номенклатури, класификатори и конфигурации (ако е необходимо въвеждането на	3 days	Mon 22-04-19	Thu 25-04-19	45					■										
56	Въвеждане на всички потребители, които няма да се определят с работен процес посредством системата.	3 days	Mon 22-04-19	Thu 25-04-19	45					■										

Project: Danriss Schedule v1
Date: Mon 25-06-18

Task		Inactive Summary		External Tasks	
Split		Manual Task		External Milestone	
Milestone		Duration-only		Deadline	
Summary		Manual Summary Rollup		Progress	
Project Summary		Manual Summary		Manual Progress	
Inactive Task		Start-only			
Inactive Milestone		Finish-only			

000045

ID	Task Name	Duration	Start	Finish	Predecessor	2019				2020				2021				
						Q2	Q3	Q4	Q1	Q2	Q3	Q4	Q1	Q2	Q3	Q4		
78	Одобрение от Възложителя	1 day	Mon 27-08-18	Mon 27-08-18	77													
79	Междинен доклад Етап 1	0 days	Mon 01-10-18	Mon 01-10-18	2													
80	Одобрение от Възложителя	5 days	Mon 01-10-18	Fri 05-10-18	79													
81	Междинен доклад Етап 2	0 days	Fri 29-03-19	Fri 29-03-19	21													
82	Одобрение от Възложителя	5 days	Mon 01-04-19	Fri 05-04-19	81													
83	Междинен доклад Етап 3	0 days	Tue 30-04-19	Tue 30-04-19	44													
84	Одобрение от Възложителя	5 days	Tue 30-04-19	Tue 07-05-19	83													
85	Окончателен доклад	0 days	Tue 30-04-19	Tue 30-04-19	44													
86	Одобрение от Възложителя	5 days	Tue 30-04-19	Tue 07-05-19	85													



Handwritten initials 'NW' on the left margin.

Handwritten initials 'NW' on the right margin.

Project: Danriss Schedule v1 Date: Mon 25-06-18	Task		Inactive Summary		External Tasks	
	Split		Manual Task		External Milestone	
	Milestone		Duration-only		Deadline	
	Summary		Manual Summary Rollup		Progress	
	Project Summary		Manual Summary		Manual Progress	
	Inactive Task		Start-only			
	Inactive Milestone		Finish-only			

000247

Handwritten signature or initials at the bottom right.