Система за електронна търговия Balkan Bay (BBay)

Главен план за разработка на софтуерен проект

Версия 1.3

История на промените

|  |  |  |  |
| --- | --- | --- | --- |
| **Дата** | **Версия** | **Описание** | **Автор** |
| 2014.01.26 | 1.0 | Първоначална версия | Михаил Радков |
| 2014.02.04 | 1.1 | Допълване на документа | Михаил Радков |
| 2014.02.08 | 1.2 | Редакция на документа | Михаил Радков |
| 2014.03.24 | 1.3 | Редакция на документа | Михаил Радков |

Съдържание

[1. Въведение 4](#_Toc383457787)

[1.1 Предназначение на документа 4](#_Toc383457788)

[1.2 Дефиниции, акроними, иабревиатури 4](#_Toc383457790)

[1.3 Препратки 4](#_Toc383457791)

[2. Обзор на проекта 4](#_Toc383457792)

[Предназначение, обхват и цели на проекта 4](#_Toc383457793)

[2.1 4](#_Toc383457795)

[2.2 Предположения и ограничения 4](#_Toc383457796)

[3. Организация на проекта 4](#_Toc383457798)

[3.1 Организационна структура 4](#_Toc383457799)

[3.2 Външни интерфейси 5](#_Toc383457800)

[3.3 Роли и отговорности 5](#_Toc383457802)

[4. Управляващ процес 6](#_Toc383457804)

[4.1 План на проекта 6](#_Toc383457805)

[4.1.1 Методологии 6](#_Toc383457806)

[4.1.2 План на фазите 7](#_Toc383457807)

[4.1.3 Цели на итерация 7](#_Toc383457808)

[4.1.4 Издания 7](#_Toc383457809)

[4.1.5 План за обучение 7](#_Toc383457810)

[4.1.6 Бюджет 8](#_Toc383457811)

[4.1.7 График на проекта 8](#_Toc383457812)

[4.2 Планове за итерации 8](#_Toc383457813)

[4.3 Мониторинг и контролиране на проекта 8](#_Toc383457820)

[4.3.1 План за контролиране на графика 8](#_Toc383457821)

[4.3.2 План за управление на качеството 8](#_Toc383457822)

[4.3.3 План за управление на риска 8](#_Toc383457823)

[4.4 План за отчитане 8](#_Toc383457824)

[4.4.1 Таблица на отчетите 8](#_Toc383457825)

[4.4.2 Легенда на таблицата с отчети 9](#_Toc383457826)

[4.4.3 Описание на отчетните документи 10](#_Toc383457827)

[5. Планове 15](#_Toc383457828)

[5.1 План за приключване 15](#_Toc383457829)

[6. Анекс 15](#_Toc383457830)

Главен план за разработкана софтуерен проект

# Въведение

## Предназначение на документа

Главният план за разработка на софтуерен проект е предназначен изпълнителя да запознае възложителя със следната информация:

* организацията на екипа за изпълнение на проекта;
* ролите и отговорностите на участниците в проекта;
* методологията за управление на проекта и фазите, на които се разделя;
* ключовите дати, идентифицирани в рамките на проекта;
* отчетните материали, които ще бъдат изготвени в рамките на проекта;
* издания на софтуерния продукт

## Дефиниции, акроними, иабревиатури

Намират се в документа „Речник“.

## Препратки

Препратките са описани в следната таблица:

|  |  |
| --- | --- |
| График на проекта | 1-I1-140202-Project Plan-v1-3.mpp |
| План за управление на качеството | 2-I1-140202-Quality Assurance Plan-1-1.docx |
| Рискове | 3-I1-140202-Risks-v1-2.docx |
| Речник | 5-I1-140202-Glossary-1-1.docx |
| Визия | 6-140202-Vision-1-2.docx |

# Обзор на проекта

## Предназначение, обхват и цели на проекта

Подробно описание може да се намери в документа „Визия“.

## Предположения и ограничения

Ограничения в проекта са откъм личния състав – само четирима души са на разположение катовсеки член на екипа има основна роля и няколко второстепенни.

Всеки от членовете на екипа има задължения към други дейности, които не се отнасят към текущия проект и поради това е невъзможно съставянето на точен график на изпълнение на проекта.

Всеки от членовете на екипа има ключова роля, и при невъзможност за изпълнение на отговорностите свързани към нея е възможно да блокира работния процес на останалите.

## Еволюция на софтуерния план

По време на изпълнение на даден проект, предвидено или не се налага да се редактира главния план за разаработка.

При възникване на нужда, главния план за разработка ще бъде редактиран, като се отчете каква е промяната и на коя дата се е случила в секцията „История на промените“.

# Организация на проекта

## Организационна структура

Организационната структура е показана на следната диаграма, където освен името на члена от екипа са указани и неговите роли, които изпълнява. Идеята на структурата е взаимствана от SCRUM където всички членове са равни.

Проектът е поставен като задача по „Практикум 1“ от специалността „Софутерно Инженерство“. Ръководител на предмета е гл. ас. Иван Станев, който влиза в ролята на възложител на проекта.

Комуникацията с възложителя ще се осъществява в предвидените часове по дисциплината, а при нужда от допълнителни срещичрез договаряне за друго време и място.

## Външни интерфейси

Разработваната ситема ще комуникира с Paypal като система за разплащане. Предвижда се проучване на възможността за комуникация и с други такива системи, за да не бъдат задължавани потребителите да ползват една единствена.

## Роли и отговорности

В следващата таблица са описани ролите, които членовете на екипа изпълняват. Всяка роля е написана с българското си име, английското и английското си съкращение.Всички членове на екипа ще изпълняват и ролите Разработчик и Тестер.

| **Роля** | **Описание** |
| --- | --- |
| Ръководител на екип Project Manager (PM) | Отговаря за разпределението на ресурсите, определя приоритетите, координира обратната връзка с клиента като цяло цели да държи екипа фокусиран към крайната цел. Определя практиките, които ще осигурят качеството на целостта и качеството на предадените с проекта артефакти; |
| Инженер по изискванията  Requirements Engineer (RE) | Води и координира събирането на изискванията и изграждането на случаите на използване чрез очертаване на функционалността и границите на системата. |
| Софтуерен Архитект  Software Architecture (SA) | Води и координира техническите действия и технически насочените артефакти по време на проекта. Софтуерния архитект определя главната структура на като съвкупност от различни погледни над системата чрез архитектурния документ. За разлика от останалите роли, цели широко познаване на областта, а не дълбоко; |
| Разработчик  Developer (DEV) | Разработчикът отговаря за написването, документирането и unit тестване на системите компоненти. Разработените от тази роля артефакти трябва да съответстват на стандартите, наложени за проекта. |
| Тестър  Tester (UT) | Извършва функционално тестване според тестовия модел и тестовия план като обобщава резултата и описва разминаванията и проблемите. Тестерът е отговорен за основните дейности по тестване на системата и верифициране на нужното й качество. |
| Бизнес Анализатор  Bussiness Analyst (BA) | Отговаря за изучаването на процесите в организацията, анализирането им и прилагането им в процеса на разработка. |
| Технически документатор  Technical Writer (TW) | Доставя материали за крайния клиент. (обучителни, помощни текстове, бележки и др.) |
| Тест Мениджър  Test Manager (TM) | Отговорен за целия процес по тестване и качество на тестовете. |
| Интеграционен тестър  Integration Tester (IT) | Планира и изпълнява билдове на готова функционалност. |
| Дизайн мениджър  Design Manager (DM) | Определя част от системата по изискванията и архитектурата. |
| Имплементационен Мениджър  Implementation Manager (IM) | Отговорен за разработвания софтуер и тестовете към него. Свръзва разработени модули. |
| Системен администратор  System Administrator (SysA) | Отговорен за целия хардуер, софтуер и инфраструктура необходими при разработването на продукта. Прави инсталации, конфигурации и резервни копия. |
| Отговорник по качеството  Quality Manager(QM) | Отговаря за осигуряването на качеството на софтуерния продукт. Той описва задачите и отговорностите, които ще се извършват от екипа, за да се осигури качеството на софтуерния продукт. |
| Продуктен Собственик Product owner | Продуктния собственик е отговорното лице за това организацията да добавя стойност (към продукти и услуги) за клиентите. Той е гласа на клиента и прави всякакви неща свързани с негоили е дори самия клиент |
| Scrum Господар Scrum master | Scrum Господарят е отговорникът за това да бъдат премахнати пречките пред това екипа да изпълни договорените за спринта задачи и да постигне желаните за спринта цели. Той не е лидер на екипа, а нещо като служещ на екипа лидер. Следи за изпълнение на правилата и за това нещата да се случват по концепциите на SCRUM. SCRUM господарят се грижи за това екипът да не бъде разсейван със странични фактори и за това всичко да е подчинено на целтие на спринта. |

# Управляващ процес

## План на проекта

### Методологии

Методологията, която ще се използва при реализирането на този проект е RUP (Rational Unified Process). RUP е постепенен процес, при който цялостният проект е разделен на фази и итерации. За всяка итерация човек трябва да достави изпълнимия софтуер, който е доказуем и проверими срещу изисквания на проекта и потребителските случаи.

Освен RUP, в итерациитеот детайлизиране нататък, ще бъде приложена и методологията SCRUM. Една от идеите на SCRUM е необходимата работа, която трябва да бъде свършена, да бъде разделена на подзадачи, всяка от които представлява малка, но готова функционалност. Всяка задача се приоритизираи тези с най-висок приоритет се разработват първи.

### План на фазите

RUPопределя четирифази при разработката на софтуерни проекти. Във всяка фаза екипът по изпълнението се фокусирана конкретен аспект от проектаисвързаните с негоредицамеждинни точки (milestones). Тезимеждинни точкипомагат наръководителя на проекта да направи оценка на напредъка на и се гарантира, че проектът ще доведе донеобходимифункциис нужното качество.Фазите според предписанията на RUP и спрямо фокуса им са както следва:

* Планиране (Inception) — Фокуса на тази фаза е да се разбере правилно обхвата на проекта;
* Детайлизиране (Elaboration) — Трябва да се създаде архитектурата на системата, заедно с изискванията към нея;
* Изграждане (Construction) — Основната дейност е разработването на проектирания в предната фаза софтуер;
* Предаване (Transition) — Софтуерът трябва да бъде внедрен и предаден на клиентите по време на тази фаза.

### Цели на итерация

* I1 – Планиране на системата
* E1- Събиране на изисквания
* E2 –Архитектура на системата
* E3 – Прототип на системата и дизайн модел
* C1 – Модули потребители, покупки и интерфейс
* C2 – Модули сигурност, плащания и отчети
* C3 – Модули за администриране, оправяне на бъгове и тестване
* T1 – Завършен продукт

Подробно описание за всяка итерация може да се намери в документа „График на проекта“ както и в детайлните документи за всяка итерация.

### Издания

Всяко издание е за края на съответната итерация.

* E3 - Прототип на системата
* C1 – Демо на функционалността разработена в тази итерация(модул потребители, покупки и интерфейс).
* C2 - Демо на функционалността разработена в тази итерация(модул сигурност, плащания, отчети и склад).
* С3 - Демо на функционалността разработена в тази итерация(административни) и предоставяне на бета версия на системата
* Т1 – Завършена система

Освен горе упоменатите издания, възложителя има право да изиска демонстрация по всяко време натекущата функционалност.

### План за обучение

Планът за обучение включва:

* След установяване на използваните технологии, на всеки член от екипа, за който се прецени, че е нужно ще бъде проведен курс.
* Курсовете ще бъдат кратки, но подробни и ще се провеждат от членове на екипа, които са по-опитни в съответната технология.

По време на разработката ще бъдат съставени упътвания за потребители и администратори.

### Бюджет

Подходът за разработка ще е с използване на продукти с отворен код, затова няма да е необходимо никакво закупуване на софтуер. Машини за разработка ще бъдат предоставени от РУ.

Проектът се реазлизира като част от дисциплината "Практикум 1" от специалността "Софтуерно инженерство". Изпълнителя (екипът) не получава никакво заплащане. Поради това е съставен бюджет с учебна цел.

Бюджетъте подробно описан в документа “Бюджет”.

### График на проекта

Графика на проекта е описан в документа „График на проекта“, който е съставен чрез MS Project.

## Планове за итерации

Всяка итерация е описана в документа „График на проекта“.

## Мониторинг и контролиране на проекта

### План за контролиране на графика

Всяка стъпка от проекта ще бъде описана като една или повече задачи в предоставената система JIRAот РУ „Ангел Кънчев“. Системата позволява създаване на различни по тип задачи, в които се описва колко време е отделено за тях и какво е свършено за това време. Притежава функционалност за следене на прогреса и генериране на отчети по всяко време.

Благодарение на това е възможно бързо и ефикасно вземане на решения свързани скоригиране на графика на проекта.

### План за управление на качеството

Планът за управление на качеството е описан в документа „План за управление на качеството“.

### План за управление на риска

Планът за управление на рисковете е описан в документа “Рискове”

## План за отчитане

### Таблица на отчетите

На таблицата по-долу е представен списък на отчетните резултати, които са предвидени да бъдат предавани по проекта и какво е очакваното им състояние в края на всяка итерация. За всеки документ е посочен отговорник.

|  | **Наименование** | **I1** | **E1** | **E2** | **E3** | **C1** | **C2** | **C3** | **T1** | **Отговорник** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | **План за разработка на софтуерния продукт** | **s** | **r** | **r** | **r** | **r** | **r** | **r** | **r** | **Михаил Радков** |
| 2 | **План за управление на качеството** | **c** |  |  |  |  |  |  |  | **Лиляна Маринова** |
| 3 | **Списък на рисковете** | **s** | **r** | **r** | **r** | **r** | **r** | **r** | **c** | **Михаил Радков** |
| 4 | **Детайлен план за итерация** | **c** | **c** | **c** | **c** | **c** | **c** | **c** | **c** | **Михаил Радков** |
| 5 | **Речник** | **e** | **e** | **e** | **s** | **r** | **r** | **r** | **c** | **Михаил Радков** |
| 6 | **Визия** | **e** | **c** |  |  |  |  |  |  | **Симеон Илиев** |
| 7 | **Бизнес модел\*** | **e** | **e** | **e** | **s** | **r** | **r** | **r** | **c** | **Малвина Макариева** |
| 8 | **Модел на потребителските случаи\*** | **e** | **e** | **s** | **r** | **r** | **r** | **r** | **c** | **Лиляна Маринова** |
| 9 | **Спецификация на допълнителните изисквания\*** | **e** | **s** | **r** | **r** | **r** | **r** | **r** | **c** | **Лиляна Маринова** |
| 10 | **Конвенции за писане на код** | **c** |  |  |  |  |  |  |  | **Малвина Макариева** |
| 11 | **Инструменти** | **e** | **s** | **r** | **r** | **r** | **r** | **r** | **c** | **Михаил Радков** |
| 12 | **Модел на данните\*** | **e** | **e** | **e** | **s** | **r** | **r** | **r** | **c** | **Лиляна Маринова** |
| 13 | **Модел на инфраструктурата\*** | **e** | **s** | **r** | **r** | **r** | **r** | **r** | **c** | **Михаил Радков** |
| 14 | **Дизайн модел\*** |  |  |  | **s** | **r** | **r** | **r** | **c** | **Симеон Илиев** |
| 15 | **Софтуерна архитектура\*** |  | **e** | **s** | **r** | **r** | **r** | **r** | **c** | **Симеон Илиев** |
| 16 | **Модел на имплементацията\*** |  |  |  |  | **s** | **s** | **s** | **c** | **Михаил Радков** |
| 17 | **Прототипи** |  |  |  | **c** | **c** | **c** | **c** | **c** | **Михаил Радков** |
| 18 | **Завършена система** |  |  |  |  | **s** | **s** | **s** | **c** | **Михаил Радков** |
| 19 | **Главен план за тестване** |  | **e** | **e** | **s** | **r** | **r** | **r** | **c** | **Малвина Макариева** |
| 20 | **Тестов модел\*** |  | **e** | **e** | **e** | **s** | **s** | **s** | **c** | **Малвина Макариева** |
| 21 | **Резултати от тестовете\*** |  |  |  |  | **c** | **c** | **c** | **c** | **Малвина Макариева** |
| 22 | **План за внедряване** |  |  |  |  | **s** | **s** | **s** | **c** | **Малвина Макариева** |
| 23 | **Материали за инсталиране, администриране, поддръжка\*** |  |  |  |  | **s** | **s** | **s** | **c** | **Симеон Илиев** |
| 24 | **План за обучение** |  |  |  |  | **s** | **c** | **c** |  | **Малвина Макариева** |
| 25 | **Материали за обучение** |  |  |  |  | **c** | **c** | **c** |  | **Лиляна Маринова** |

### Легенда на таблицата с отчети

За обозначаване на статуса на отчетните резултати през отделните итерации на проекта са използвани следните символи

|  |  |
| --- | --- |
| Код | Описание |
| e (evolved) | В процес на разработка (по резултата се работи) |
| s (stable) | Приет (утвърден) |
| r (revised) | Ревизиран (актуализиран след приемане, при необходимост) |
| c (completed) | Завършен (замразен) |

### Описание на отчетните документи

Детайлното описание на статуса за всеки отделен отчетен материал за различните модули може да се видят на следващата таблица:

|  | **Наименование** | **Описание** |
| --- | --- | --- |
| 1 | План за разработка на софтуерния продукт | Планът за разработка на софтуерния продукт представлява комплексен, интегриран артефакт, в който е съсредоточена цялата информация, необходима за управлението на проекта. Той включва редица артефакти, разработвани по време на фаза Планиране, и се поддържа през целия жизнен цикъл на проекта.  Фаза Планиране: Извършва се първоначална идентификация на фазите, техните срокове и цели. В рамките на Плана за разработка на софтуерния продукт се специфицират и съгласуват разчети на ресурсите. Обсъжда се и се залага първи вариант на Плана за приемане на продукта. Последният се уточнява през следващите итерации, когато възникват допълнителни изисквания.   Фаза Детайлизиране: Планът се актуализира и допълва така, за да обхване фази Изграждане и Предаване. |
| 2 | План за управление на качеството | ПУК е артефакт, който създава ясна визия за начините, по които ще се управлява качеството на продукта, артефактите и процесите. Част от него е Планът за прегледи и одит. В Плана за качеството са посочени и редица други артефакти, разработвани през фаза Планиране. Планът за качеството се поддържа през целия жизнен цикъл на проекта. |
| 3 | Списък на рисковете | Структуриран списък на известните и реални рискове за проекта. Рисковете са подредени в низходящ ред на значимост и към всеки от тях са привързани конкретни мерки за ограничаване на последствията или действия при непредвидени ситуации.  Фаза Планиране: Съставя се Списък на рисковете, в който са отразени предвижданите рискове за успеха на проекта. Първите рискове се идентифицират именно през фаза Планиране.  Фаза Детайлизиране и Изграждане: Списъкът на рисковете се преразглежда и актуализира за всяка итерация. |
| 4 | Детайлен план за итерация | Включва всички дейности и резултати за конкретната итерация, които са дефинирани последователно във времето и са обвързани със съответните ресурси.  Планът за итерация N се предава в края на итерация N-1. |
| 5 | Речник | В Речника се дефинират всички важни термини, използвани за целите на проекта. |
| 6 | Визия | Във Визията е заложено виждането на участниците в проекта относно продукта (основни нужди и характеристики). Така тя става "договорна база" за по-подробно дефиниране на техническите изисквания.  Фаза Планиране: Основните изисквания, характеристики и ограничения на проекта се документират.  Фаза Детайлизиране: Визията се прецизира на база новата информация, получена през тази фаза, създавайки по този начин устойчиво разбиране за най-критичните потребителски случаи, които ще са в основата на архитектурните и планировъчни решения. |
| 7 | Бизнес модел | Бизнес моделът описва процесите вътре в организацията и взаимодействията й с външни страни, като клиенти или партньори. Освен това, Бизнес моделът описва начина, по който се изпълняват бизнес потребителски случаи. |
| 8 | Модел на потребителските случаи | Моделът на потребителските случаи представя предвижданите функции и среда на системата и отразява нейните функционални спецификации Този модел е важен входен артефакт за дейностите по анализ, дизайн и тестване.   Фаза Планиране: Идентифицират се важните актьори и потребителски случаи, както и последователностите от събития в рамките на най-важните потребителски случаи.  Фаза Детайлизиране: Моделът на потребителските случаи е почти завършен (на около 80%) – всички потребителски случаи са идентифицирани при проучването на модела, всички актьори също са идентифицирани и повечето потребителски случаи са описани (установени са изискванията).  Фаза Изграждане: Моделът се актуализира с нови потребителски случаи, ако такива са установени през фаза Изграждане. |
| 9 | Спецификация на допълнителните изисквания | Спецификацията на допълнителните изисквания отразява тези изисквания към системата, които не са ясно прихванати в отделните потребителски случаи на Модела на потребителските случаи. Тук се включват правни и нормативни изисквания, както и приложни стандарти. Необходимо е да се изведат характеристиките на качеството на системата, в това число изискванията по отношение на използваемост, надеждност, производителност и съпроводимост. Определят се и други изисквания, например по отношение на операционната система, работната среда и съвместимостта, както и ограниченията по отношение на дизайна.  Фаза Детайлизиране: Допълнителните изисквания по отношение на нефункционалните характеристики биват документирани и съгласувани.  Фаза Изграждане: Спецификацията на допълнителните изисквания се актуализира с нови изисквания, ако такива бъдат установени по време на фаза Изграждане. |
| 10 | Конвенции за писане на код | Този документ съдържа набор от стандарти, конвенции и указания за писане на стабилен Java код. Тези стандарти, конвенции и указания почиват върху солидни, доказани принципи за софтуерно инженерство, водещи до създаването на код, който е разбираем, удобен за поддържане и податлив на усъвършенстване. |
| 11 | Инструменти | Фаза Планиране: Подбират се всички инструменти, необходими за поддържане на проекта. Инсталират се инструментите, необходими за работата през фаза Планиране.  Фаза Детайлизиране: Среда за разработка, включително необходимите на екипа по Изграждане процеси, инструменти и средства за автоматизация, се (разработва и) изгражда през тази фаза. |
| 12 | Модел на данните | Моделът на данните е подмножество на имплементационния модел, което описва логическия и физически вид на постоянните (персистентни) данни в системата. Той включва и видовете поведения в базата данни, например записани процедури, активатори, ограничения и др.  Фаза Детайлизиране: Дефиниране и залагане на модела. Извършва се дефиниране и преглед на основните елементи от модела на данните (напр. важни обекти, взаимозависимости, таблици).  Фаза Изграждане: Моделът са актуализира с всички елементи, необходими за поддържане на постоянната (персистентна) имплементация (например таблици, индекси, конверсии между обектни и релационни форми и др.). |
| 13 | Модел на инфраструктурата | Моделът на инфраструктурата показва конфигурацията на обработващите възли в работни условия, комуникационните връзки между тези възли, както и заложените в тях инстанции на компонентите и обекти. |
| 14 | Дизайн модел | Дизайн моделът е обектен модел, който описва реализацията на потребителските случаи и служи за извеждане на Модела на имплементацията и неговия програмен код. Дизайн моделът е важен входен артефакт за дейностите по имплементация и тестване.  Фаза Детайлизиране: Дефиниране и подготвяне на базова версия.  Фаза Изграждане: Моделът (и всички съставни артефакти) се актуализира(т) с нови елементи на дизайна, установени при реализацията на всички изисквания. |
| 15 | Софтуерна архитектура | Документът Софтуерна архитектура представя комплексен архитектурен изглед на системата, използвайки за целта редица различни архитектурни разрези, показващи отделни нейни аспекти.  Фаза Детайлизиране: Създава се основна версия, включвайки подробни описания на архитектурно значимите потребителски случаи (разрез по потребителски случаи), идентификация на основните механизми и елементи на дизайна (логически разрез), плюс изглед по процеси и изглед по внедряване (в съответствие с Модела на внедряване), ако системата е разпределена. |
| 16 | Модел на имплементацията | Моделът на имплементацията събира на едно място компонентите и съдържащите ги имплементационни подсистеми. Компонентите включват както тези, които подлежат на предаване като отчетни резултати (например изпълнимите компоненти), така и тези, от които се извеждат предаваните компоненти (например файлове с програмен код).  Фаза Детайлизиране: Създава се първоначалната структура. Идентифицират се основните компоненти и се разработват прототипи на тези компоненти. Моделът на имплементацията е набор от компоненти и съдържащите ги имплементационни подсистеми. Компонентите включват както тези, които подлежат на предаване като отчетни резултати (например изпълнимите компоненти), така и тези, от които се извеждат предаваните компоненти (например файлове с програмен код).  Фаза Изграждане: Моделът (и всички съставни артефакти, включително компонентите) се доразвива(т), надграждайки върху създаденото през фаза Детайлизиране. Към края на фаза Изграждане всички компоненти са вече създадени. |
| 17 | Прототипи | Фаза Планиране: Един или няколко прототипа, разработени с цел доказване изпълнимостта на концепцията, поддържане на Визията и решаване на най-специфичните рискове.  Фаза Детайлизиране: Създават се един или няколко изпълними архитектурни прототипа за изследване на критичната функционалност и архитектурно значимите сценарии. |
| 18 | Завършена система | Изпълнимата система като такава, готова за стартиране на бета тестове. |
| 19 | Главен план за тестване | Планът за тестване определя стратегиите за разработване и изпълнение на тестовете, както и необходимите ресурси. |
| 20 | Тестов модел | Тестовият модел обяснява какво и как ще бъде тествано. Той представлява разрез на моделите на бизнес процесите, на дизайна и имплементацията, с който са описани самите тестове и тези аспекти на тестваните артефакти, които са от значение за тестовия процес.  Тестовият модел включва набор от тестови случаи, тестови процедури и тестови скриптове, както и описание на очакваните резултати от тестовете и на взаимовръзките помежду им. Трябва да съдържа също таблици на съответствието бизнес процеси – системни потребителски случаи – компоненти на системата - тестови случаи и тестови сценарии.  Фаза Изграждане: Извършва се дизайн и разработване на тестове за проверка на изпълнимите версии, създавани през фаза Изграждане.  Таблиците на съответствието се поддържат в актуално състояние през целия период на изпълнение на проекта. |
| 21 | Резултати от тестовете | Този артефакт представлява архив от данни, получени при провеждането на тестовете. Използва се при изчисляването на различни ключови метрики на тестовете.  Резултатите от тестовете трябва да са придружени с таблица обобщаваща резултатите от изпълнението на тестовете, включваща и информация за дефектите, които са регистрирани в резултат на неуспешно изпълнение. |
| 22 | План за внедряване | Планът за внедряване описва набора от задачи по инсталиране и тестване на разработения продукт с оглед успешното му предаване на потребителя.  Фаза Изграждане: Разработване, съгласуване и залагане на първата версия. |
| 23 | Материали за инсталиране, администриране, поддръжка | Инсталационните артефакти включват софтуера и документираните инструкции (и скриптове) за генериране на версии от програмния код, инсталация, конфигурация и миграция, Ръководство за потребителя и Ръководство за администратора и др. |
| 24 | План за обучение | Включва анализ на потребностите от обучение и стратегия за провеждане на обучението. |
| 25 | Материали за обучение | Това са материали, които се използват при провеждането на курсове или програми за обучение, предназначени да подпомогнат бенефициента при използването, експлоатацията и/или обслужването на системата.  Фази Детайлизиране и Изграждане: Изготвят се Ръководства на потребителя и други материали за обучение. Това са предварителни варианти, базирани на потребителските случаи. Такива са необходими, когато системата осъществява интензивен интерфейс с потребителя.  Фаза Предаване: Материалите се привеждат в завършен вид така, че клиентът да може самостоятелно да използва и обслужва продукта. |

# Планове

## План за приключване

В края на всяка итерация се предават определените документи на възложителя, който ги преглежда и преценява дали да се продължи към следваща итерация или да ги върне за ревизия.

В итерация „Предаване“, изпълнителят предава на възложителя готовия софтуерен продукт заедно с цялата необходима документация. Ако възложителят определи, че целите на проектирането и реализацията на продукта са реализирани и удовлетворяващи, то проекта се приключва.

# Анекс

|  |  |
| --- | --- |
| Бюджет | 0-I1-140202-Budjet.xls |