**Единна деловодна система на държавна администрация (**Unified Case Management System for the National Administration, UCMS)

**План за разработка на продукта**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Версия** | **Дата** | **Автор** | **Описание** | **Стр.** |
| 1.0 | 28.09.2013 | И. Русев | Създаване на документа |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

1. **Въведение**

## **Съдържание**

**1.** **Въведение**

*[1.1](#h.30j0zll)**[Съдържание](#h.30j0zll)*

*[1.2](#h.1fob9te)**[Съкращения](#h.1fob9te)*

*[1.3](#h.3znysh7)**[Цел на документа](#h.3znysh7)*

**[2.](#h.2et92p0)****[Въведение в проекта](#h.2et92p0)**

*2.1* *Основни цели и обхват*

*2.2* *Организация на проекта*

[2.2.1 Организация на бенефициента](#h.1t3h5sf)

[2.2.2 Организация и организационна структура на изпълнителя](#h.4d34og8)

*2.3* *Роли и отговорности на екипа на изпълнителя*

[2.3.1 Роли по RUP](#h.147n2zr)

[2.3.2 Профили на членовете на екипа](#h.3o7alnk)

*[2.4](#h.23ckvvd)**[Начин на провеждане на проекта](#h.23ckvvd)*

[2.4.1 Итеративна разработка](#h.ihv636)

[2.4.2 Междинни точки](#h.32hioqz)

*2.5* *Комуникация между екип едно и РУ*

*[2.6](#h.41mghml)**[Следене напредъка по проекта](#h.41mghml)*

*[2.7](#h.2grqrue)**[Управление на рисковете](#h.2grqrue)*

*[2.8](#h.vx1227)**[Външни зависимости](#h.vx1227)*

*[2.9](#h.3fwokq0)**[Управление на разработката и приоритизация](#h.3fwokq0)*

*[2.10](#h.1v1yuxt)**[Процедура по искане на промяна](#h.1v1yuxt)*

[2.10.1 Текущ проблем](#h.4f1mdlm)

[2.10.2 Предложена промяна](#h.2u6wntf)

[2.10.3 Решение](#h.19c6y18)

**[3.](#h.3tbugp1)****[График на проекта](#h.3tbugp1)**

*[3.1](#h.28h4qwu)**[Оценка на усилието и размера на проекта](#h.28h4qwu)*

*[3.2](#h.nmf14n)**[График](#h.nmf14n)*

*[3.3](#h.37m2jsg)**[Действия по приключването на итерация](#h.37m2jsg)*

*[3.4](#h.1mrcu09)**[Критерии за оценка на итерация](#h.1mrcu09)*

*[3.5](#h.46r0co2)**[Общ преглед на функционалностите на системата](#h.46r0co2)*

*[3.6](#h.2lwamvv)**[Детайлен план за итерация Е1](#h.2lwamvv)*

*[3.7](#h.111kx3o)**[Детайлен план за итерация C1](#h.111kx3o)*

*[3.8](#h.3l18frh)**[Детайлен план за итерация C2](#h.3l18frh)*

**[4.](#h.206ipza)****[Човешки ресурси](#h.206ipza)**

**5.** **Отчетни резултати**

*5.1* *Итерации и отчетни резултати*

*5.2* *Описание на отчетните резултати*

**6.** **Бюджет**

*[6.1](#h.3cqmetx)**[Бюджет за човешки труд](#h.3cqmetx)*

*[6.2](#h.4bvk7pj)**[Бюджет за софтуер и консумативи](#h.4bvk7pj)*

## **Съкращения**

Съкращенията се намират в Glossary.doc

## **Цел на документа**

Този документ представя:

* организацията на екипa 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;

1. **Въведение в проекта**

## Основни цели и обхват

Основните цели на проекта са проектирането и програмното разработване на единна деловодна система за целите на държавна администрация. Системата ще служи за обработката и съхраняването на документация от различни деловодни системи.

## **Организация на проекта**

В точката е описана организацията на проекта от гледна точка на организацията и стейкхолдърите.

Стейкхолдърите на проекта са определени по препоръките на “PRINCE2 Project Management Roles”.

**Ръководител на проекта от страна на стейкохолдърите**: организирането и контролирането на проекта означава, че трябва да човек занимаващ се с организацията и контролирането. Ръководителя на проекта избира кои хора да работят по проекта и е отговорен, че работата ще се свърши добре и навреме. За проекта по практикум 3 ръководителя на проекта от страна на стейкхолдърите е И. Станев.

**Клиент**: Е организацията, която е поръчала и получава разработения продукт. Университетът се явява клиент и запазва правото да използва, разработения от екипа проект.

Потребител (Senior User) е човекът, който ще използва продукта. За организацията на университета това се явяват служителите от катедра „ИИТ”

**Снабдител(Supplier)** са хората, които имат нужното знание и могат да отпуснат необходимите ресурси за проекта. Поради специфичния характер на проекта (като част от университетска дисциплина) снабдителя са водещите на дисциплината. Те разпределят екипите и отпускат нужното време и ресурси за изпълнението. Те са от екипа на възложителя, а именно ръководителите на дисциплината „практикум 3” към магистратурата „Софтуерно инженерство“.

Следва графично представяне на организацията на проекта:

### *Организация на бенефициента*

Поръчителят или бенефициента се състои от: ръководителя на проекта от страна на стейкхолдърите, клиента, потребителя и снабдителя. Изпълнителят предава разработените артефакти директно на член екипа на бенефициента (или целия екип). При неясноти, изникнали в процеса на разработка в зависимост от областта на въпроса човек от екипа на извършителя се свързва с поръчителя.

### *Организация и организационна структура на изпълнителя*

Екипът по изпълнението използва проектна организационна структура. Тя може да се нарече още временна организационна структура,  защото се създава за решаването на конкретна задача.   Смисълът   на   нейното   създаване   се   състои   във формиране на екип от сътрудници на организацията (в случая студенти от университета) за работа над сложен проект с фиксиран срок, качество и разходи. Когато проектът бъде завършен, екипът се разпуска. Сътрудниците започват да работят по нов проект или се завръщат в стационарния отдел, или напускат организацията.

## 



### 



### 

* нея

## **Роли и отговорности на екипа** **на изпълнителя**

Проекта се извършва от екип от 7 човека от страна на изпълнителя. Всеки от хората има определена роля и отговаря за разработването на част от проекта. Координацията се поема от ръководител на проекта (PM). Той води комуникацията с клиента на управленско ниво, планира детайлно задачите към отделните членове на екипа и определя сроковете за изпълнение в съгласие с възложителя.

Екипа носи отговорността за успеха на проекта и неговото завършване качествено и в срок. Поради организацията на проекта „Единна деловодна система за държавна администрация“ се налага изпълнителите да заемат повече от една роля. Всеки член на екипа на изпълнителя има основна роля и една или няколко второстепенни. В отделни фази в зависимост от сроковете и поставените задачи може да се наложи поставянето на един, няколко или всички членове на екипа в една и съща роля, различна от първостепенната и второстепенната.

### *Роли по* RUP

Практикум 3 се води по методологията на RUP, затова на всеки човек от екипа на изпълнителя се дава определен профил, който съдържа в себе си една или няколко RUP роли. Следва описание на RUP ролите, които ще бъдат разпределени в профили.

|  |  |  |
| --- | --- | --- |
| Българско наименование | Английско наименование | Отговорности |
| Ръководител на екип | Project Manager | 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 цялостта и качеството на предадените с проекта артефакти; |
| Софтуерен Архитект | Software Architect | 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; |
| Инструментален специалист | Tools specialist | Специалист, познаващ инструментите използвани по времето на разработката, както тези за самото ръководене на проекта, така и технологиите в самата разработка. |
| Събирач на изисквания | Requirements Specifier | Води и координира събирането на изискванията и изграждането на случаите на използване чрез очертаване на функционалността и границите на системата.  Отговаря за:  • координиране на всички аналитични дейности в рамките на проекта;  • прехвърлянето на събраните знания към програмистите, тестерите и архитектите на системата;  • представлява нуждите на потребителя пред екипите на изпълнителя на проекта;  • поддържа обхвата на проекта;  • дава становища на ръководителя на екип по отношение обхвата на проекта, искания за промяна и въпроси;  • изработването и доставката на моделите на системата на екипите по изграждане;  • информиране на тест мениджъра за възможните промени на определени функции. |
| Разработчик | Implementer | Разработчикът отговаря за написването, документирането и unit тестване на системите компоненти. Разработените от тази роля артефакти трябва да съответстват на стандартите, наложени за проекта. |
|  |  |  |
| Buisness Process Analyst | Анализатор на бизнес процеси. | Отговаря за изучаването на процесите в организацията, анализирането им и прилагането им в процеса на раработка. |
| Системент анализатор | System Analyst | Управлява процеса по избиране и моделиране на потребителски случаи чрез избирайки ключовите за системата функционалности. |
| Тестер | Tester | Извършва функционално тестване според тестовия модел и тестовия план като обощава резултата и описва разминаванията и проблемите. Тестерът е отговорен за основните дейности по тестване на системата и верифициране на нужното й качество. |
| Организатор на обучения | Course Organizer | Грижи се за организацията по провеждането на обучения по време на процеса по внедряване на системата. |
| Технически документатор | Tech Writer | Доставя материали за крайния клиент. (обучителни, помощни текстове, бележки и др.) |
| Тест дизайнер | Test Designer | Отговорен е за определяне подхода на тестване и успешното му прилагане. Ролята включва идентифициране на правилните техники, тулове и ресурси. |
| Тест мениджър | Test Manager | Отговаря за успешното провеждане на тестовете. |
| Интегратор | Integrator | Сглобява създадените и тествани от него и други разработчици функционалности в по-големи модули, така че да се получи стабилен билд; |

### *Профили на членовете на екипа*

Профилите на хората от екипа са както следва:

#### Профил: „Ръководител на екипа“, изпълняван от Иво Русев.

* Ръководител на екип (Project Manager)
* Софтуерен архитект (Software Architect);
* Интегратор (Integrator);
* Инструментален специалист (Tools specialist).

#### Профил „Бизнес анализ и инженер по качество“

Изпълняван от Искрен Папазов и Бояна Димова. Екипът по качеството отговаря за събиране на изискванията, стабилността на системата и качеството на разработения продукт.

Включва следните RUP роли:

* Анализатор на бизнес процеси (Buisness Process Analyst)
* Системент анализатор (System Analyst)
* Събирач на изисквания (Requirements Specifier)
* Тест мениджър (Test Manager)
* Тест дизайнер (Test Designer)
* Тестер (Tester)

#### Профил „Бизнес анализ и разработка“

Изпълняван от Антоний Кунчев. Разработчик с опит в събирането на изисквания, който може да следи, че проектираните функционалности ще могат да бъдат реализирани в рамките на зададените срокове:

Включва следните RUP роли:

* Събирач на изисквания (Requirements Specifier)
* Интегратор (Integrator)

#### Профил „Разработка“

Изпълняван от Виктор Костадинов, Станислав Стефанов. Разработчиците, отговарящи за разработването на функционалностите. Включва както разрабочици с повече така и такива с по-мало опит:

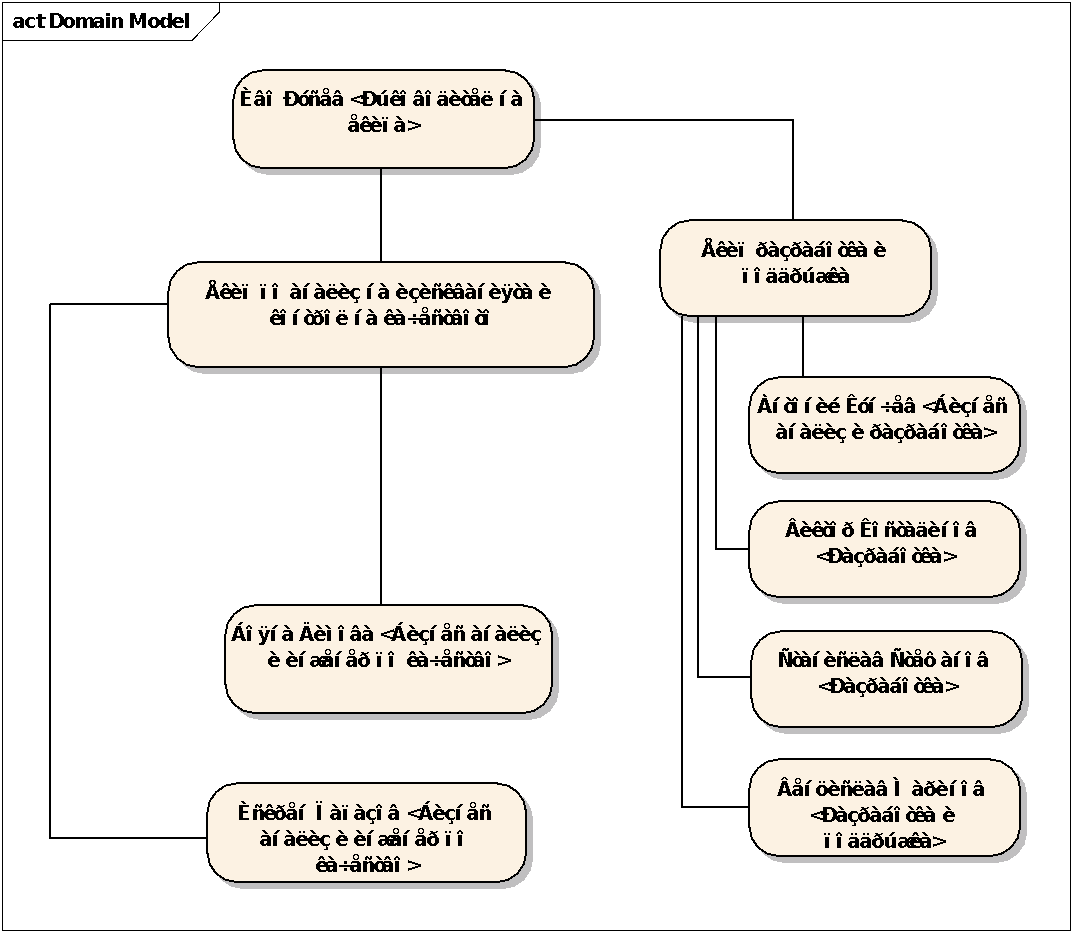
* Разработчик
* Интегратор

#### Профил „разработка и поддръжка“

Изпълняван от Венцислав Маринов. Разработчик отговарящ освен за разработването на функционалностите по системата и за документацията свързана с ръководства и материали:

* Разработчик
* Deployment manager
* Организатор на обучения (Course Organizer)
* Технически документатор

Долу е представена графично организационната структура на «Екип 2»:



Такава организация ще осигури най-ефективното използване на ресурсите по проекта.

Екип бизнес анализ отговаря за събирането на изискванията и създаването на основните процеси, които трябва да бъдат реализирани. Отговаря за документите, свързани е проектирането на системата като например документа с изискванията към софтуерната система, потребителските случаи и др.

Екип „Разработка и поддръжка“ отговаря за разработването на системата и поддържането на средата за разработка. Създават технически насочените документи като например софтуерната архитектура, дизайн модела и др.

## **Начин на провеждане на проекта**

Методиката за изпълнение на проекта е по RUP, както е наложено от дисциплината "Практикум 3". Проектите по RUP имат две основни характеристики:

* Са итеративни;
* Прогреса на проекта се измерва спрямо предварително междинни точки от изпълнението.

### *Итеративна разработка*

По-голямата част от проектите по RUP, са по дефиниция, итеративни. RUP е постепенен процес, при който цялостният проект е разделени на фази и повторения. Итерациите са риск задвижвани - това е, ориентирана към намаляване на рисковете. Всяка итерация човек трябва да достави изпълнимия софтуер, който е доказуем и проверими срещу изисквания на проекта и потребителските случаи.

Ръководителят на проекта използва итерация планове за управление на проекта. Като цяло работата, която попада извън плана за итерация не трябва да бъде предприета. План за итерация:

* предоставя детайлно описание на идващата работна фаза;
* дефинира въвлечените роли, нужните действия и артефактите, които трябва да се предадат;
* дефинира ясен набор от критерии, чрез които може да се оценява прогреса през итерацията и в края й;
* Дефинира начална и крайна дата, както и дати за предаване.

В контекста на итеративна разработка, основните междинни точки осигуряват основа за итерацията. Всяка такава движи проекта през определени етапи. Например, една итерация в начална фаза ще бъде структурирана около необходимостта да разбере обхвата на проекта; итерация (и) ще осигури рамката за управление на екипа, който да проучи границите на системата, последствията от едно възможно решение, както и размерът на това решение. Броят на повторения, ще зависи от това колко трудно е да се определи обхватът на проекта. Ако обхватът е много трудно да се разбере или задачите могат лесно да бъдат групирани в определени парчета, повече от едно повторение може да са необходими. Ако случаят е, че работата може да бъде свършена в една ясно определена итерация, тогава е излишно да се добавят допълнителни.  
Основните етапи, определени в RUP по необходимост са доста общи, ръководителят на проекта ще трябва да ги насочва, така че те да фокусират екипа към нуждите на проекта, по-специално неговия организационен контекст.

Освен това тъй като целта на итерацията е да се намали риска по време на итерация вниманието на екипа ще бъде насочвано не само към разрашеването на конкретни проблеми но ще обръща внимание и на други дисциплини като например софтуерната архитектура, тестване, управление на промяната и др. Менидцърът съчетава итеративен ориентиран към рисковете подход с прецизирани задачи за точно определяне на плана на итерацията.

### *Междинни точки*

RUP определя четири фази за при разработката на софтуерни проекти. Във всяка фаза екипът по изпълнението се фокусира на конкретен аспект от проекта и свързаните с него редица междинни точки (milestones). Тези междинни точки помогат на ръководителя на проекта да направи оценка на напредъка на и се гарантира, че проектът ще доведе до необходими функции с 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 със 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 в по време на тази фаза.

Фазите са допълнително уточнени в т. 3.2.

## **Комуникация между екип едно и РУ**

Комуникацията между „Екип 2” и поръчителя ще става в рамките на часовете по Практикум 3. При нужда от допълнителни срещи ръководителят на екип се свързва с член от РУ и предава получената информация на екипа.

## **Следене напредъка по проекта**

Следенето на напредъка по проекта се осъществява чрез следните основни дейности:

* В рамките на проекта се използва системата за следене на задачи jira. Всички изпълнители по проекта редовно създават и актуализират задачи, съответстващи на действията, които извършват по проекта. Задачите се коментират на ежеседмичните срещи и проблеми по проекта се отстраняват своевременно. Задачите за всяка итерация се определят преди началото в началото й, като задачата на всеки член от екипа е да направи приблизителна оценка на поставените му задачи. По този начин „Екип 2” ще може по-лесно да оцени усилието, нужно при разработка;
* Провеждане на седмични онлайн срещи и изготвяне на отчети. Всяка седмица, четвъртък от 21:30 «Екип 2» провежда срещи по изпълнението на проекта. Анализират се изпълняваните в момента задачи и прогреса по тях. Срещите се провеждат по SCRUM методологията, като в рамките на 5 минути всеки член от екипа споделя: 1. По какво е работил; 2. По какво ще работи; 3. Има ли някакви проблеми. Тази практика позволява навременното идентифициране и отстраняване на възникнали в хода на работа проблеми;
* Провеждане на мениджърски срещи при възникване на проблеми при изпълнението между ръководителя на «Екип 1» и поръчителите от РУ. В случай, че проблемите засягат пряко някой от другите членове на екипа, те се включват също в срещата. След срещите се изготвят протоколи, за да могат и останалите членове да се запознаят с взетите решения.

## **Управление на рисковете**

Идентифициране и оценяване на рисковете в проекта е важна задача по времето на изпълнение на целия проект. Артефакта, който се произведе като резултат служи за намаляване на рисковете и разработване на планове за итерации във фазите за анализ и разработка. Според RUP се препоръча да се предприемат следните стъпки:

* Да се идентифицират основните рискове, които могат да намалят вероятността екипа да разработи проект с всички нужни характеристики;
* Да се анализират и приоритетизират рисковете, оценявайки вероятността им на настъпване и последствията от тях;
* Да се идентифицират стратегии за намаляване на риска;
* Да се идентифицират стратегии за ограничаване на риска;
* Да се проверяват и ревизират рисковете в различните итерации;

Списък с първоначално идентифицираните рискове е изготвен при стартирането на проекта. Този списък ще се ревизира с изготвянето на план за итерация за фазите детайлизиране и разработка. Списъкът се намира в UCMS-Risks-List.xls

## **Външни зависимости**

При планирането е важно да се вземат предвид външни фактори, които биха могли да повлияят на изпълнението на проекта.

В конкретния „Екип 2” ще работи по UCMS, използвайки голямо количество софтуер с отворен код, поради характера на проекта. Този тип софтуер позволява намаляване на разхода при разработка, но може да доведе до усложнения, например при излизането от поддръжка на определена, използвана от екипа версия или дефекти в съществуващи в библиотеките от трети страни, които не могат да бъдат отстранени въобще или не могат да бъдат отстранени в разумен срок.

Друга външна зависимост, която трябва да бъде взета в предвид е ограничените ресурси и възможности при изграждане на развойна среда. Тази външна зависимост ще бъде обсъдена със системните администратори.

## **Управление на разработката и приоритизация**

Поради факта, че изискванията към софтуерния продукт се променят често е нужен гъвкав подход, по който да се подходи при промяна в изискванията. Поради обхвата на проекта и фиксираните срокове за управление на промените е избрана гъвкава методология по SCRUM. Методологията е както следва:

1. Взиманат се всички, планирани за разработка в рамките на спринта потребителски случаи. Списъкит от тези изисквания представлява беклога за съответния спринт. В беклога могат да се добавят и/или премахват потребителски случаи по време на спринта при изникване на нова изисквания с по-висок приоритет;
2. Оценка и приоритизиране на всяко изискване като се подреждат от висок към нисък приоритет. Изискванията с по-висок приоритет се имплементират първи;
3. Изисквания могат да бъдат оставяни за следващ спринт или при съгласуване с поръчителя премахвани;
4. Нови изисквания от възложителя трябва да бъдат обсъдени и одобрени от екипа по разработката, по-подробно в т. 2.10;
5. Задачи към членовете на екипа се разпределят от беклога.

## **Процедура по искане на промяна**

При искане на промяна се попълва „заявка за промяна“, която трябва да включва следните полета:

* Тип на промяната (ново изискване, проблем или подобрение);
* Наименование: описание с няколко думи в какво се състои промяната;
* Приоритет: до колко е критична промяната за възложителя;
* Дата на подаване: актуалната дата на искане на промяната, попълва се от изпълнителя в момента на получаването;

Заявката трябва да включва следните секции:

### *Текущ проблем*

Тази секция включва подробно описание на проблема, наложил искането на промяната, кога се наблюдава проблема. Ако е възможно да бъде изваден списък със стъпки които довеждат до проблема. Трябва да включва кратко описание как работата върху проблема ще окаже влияние над цялостното изпълнение на проекта, т.е. ресурси, които ще отнеме, влиянието му над други части на системата и срока на изпълнение.

### *Предложена промяна*

Трябва подробно да се опише предложената промяна:

* Какво точно да може да изпълнява системата в бъдеще;
* Влиянието на промяната на други части на системата;

### *Решение*

Прави се оценка от изпълнителя на функционалността, колко време ще отнеме и до какви промени по вече съществуващите функционалности ще доведе.

с На база на оценката от изпълнителя и времето за изпълнение се взема решение какво ще се реализира, в какви срокове, с какъв приоритет и за сметка на какво (при положение, че се налага намаляване на обхвата. Ако екипа има време и може да вмести промяната на база на оценката, тя се осъществява. Ако промяната не може да се вмести във времето за изпълнение и приоритета на исканата промяна надвишава приоритета на вече съществуваща, се променя обхвата на проекта като новата функционалност заменя старата, която е с най-нисък приоритет.

Решението се записва в заявката за промяна.

1. **График на проекта**

## **Оценка на усилието и размера на проекта**

По време на стартирането на всеки софтуерен проект, както и на всяка итерация, възложителите изискват оценка на нужното време и усилие. Оценката на софтуерни проекти е трудна, поради факта, че често не може да се прецени оставащата работа. В други проекти, несвързани с разработката на софтуер, често може да се види дори визуално напредъка и това колко работа остава, което от своя страна улеснява ръководителите. Основните характеристики, които правят разработването на софтуер трудно за оценка са:

* Невидимост: в софтуерната индустрия прогреса не е толкова видим както например при строенето на мост;
* Сложност: разработването на софтуер е по-сложно от другите инженерни процеси;
* Гъвкавост: софтуера лесно се променя. Това често се посочва като едно от най-големите предимства, но в същото време честите модификации могат да доведат до проблеми и забавяния;
* Уникално приложение на софтуера: другите инженерни специалности често се осланят на факта, че ако едно начинание е подобно на друго е възможно да оценим следващия проект спрямо него. Поради променяемостта на софтуера, всяко решение е уникално, което прави трудно да се даде оценка спрямо предишен проект;
* Променяща се технология: даже и на един и същи проект технологията за разработка се променя; между проекти промяната в технологията е огромна, което пречи да се даде оценка колко време ще отнеме разработката на дадена характеристика на продукта;

Оценката на усилие се подобрява по време на жизнения цикъл за разработка, когато знанията на екипа и разбирането за проекта се подобряват, но при оценка на фазите са взети в предвид горните пет насоки.

## График

Проектът е планиран по RUP методологията и е разделен на следните четири основни фази. Дейностите по отделните фази могат да бъдаъ видени във файла UCMS-Project Schedule.mpp:

* Планиране (Inception): Целта на фазата е изграждане на визията и планът за изграждане на проекта. Тук също се определя организационната схема, състава и отговорностите на членовете на екипа. В тази фаза се събират основните функционални и нефункционални изисквания.
* Детайлизиране (Elaboration): Главните цели на фазата са:
  + Изграждане на бизнес модела чрез анализ на потребителските изисквания. Изграждат се диаграми на бизнес процесите, състоянията и потребителските случаи;
  + Избор на софтуерна архитектура;
  + Изграждане на тестов модел;
  + Изграждане на модел на данните;
  + Изграждане на дизайн модел.
* Изграждане (Construction): В тази фаза основната цел е изграждането на системата. В началото се изграждат общите/базовите функции на системата. След това се изграждат функционалности, отговарящи за конкретните изисквания. Тази фаза приключва с пълно тестване на потребителските изисквания.
* Предаване (Transition): Целта на фазата е да позволи на бенефициента да извърши тестване за приемането на системата, както и да се извърши прехода към новата система.

## **Действия по приключването на итерация**

При приключването на всяка итерация се събира архив с артефакти за предаване, който включва:

* Новоразработени артефакти;
* Ревизирани артефакти

Архивът не включва артефактите, по които няма промени. Договаря се с поръчителя начин, по който ще се предава всяка итерация. Артефактите се подбират спрямо точка пет: „Отчетни резултати”

След всяка итерация (без итерацията на фаза планиране, поради прекалено скорошното започване на работата) се извършва ретроспекция. На ретроспекцията всеки член на екипа отбелязва какво му е харесало/подразнило в изминалия спринт и на базата на тази информация се избират теми, които да се обсъдят и да доведат до подобряване.

## **Критерии за оценка на итерация**

Критериите за оценка са степента на завършеност на поставените задачи и обратна връзка от поръчителя. Задачите за спринта се поставят и наблюдават в jira, а документите се ревизират и обсъждат на ежеседмичните срещи от членовете на екипа.

## **Общ преглед на функционалностите на системата**

Бизнес функционалностите, които ще бъдат разработени са разделени на групи или казано по друг начин модули, както следва:

|  |  |
| --- | --- |
| **Група функционалности** | **Функционалности** |
| Изграждане на модул "Регистрация" | * Регистрация на потребител; * Регистрация на организация. |
| Изграждане на модул "Архивиране" | * Изпращане на документ за архивиране; * Автоматично архивиране на определен период. |
| Изграждане на модул "Управление на профили" | * Задаване на права за различни потребители; * Преглед на информацията за профила; * Редактиране на информацията за профила; |
| Изграждане на модул "Помощ и въпроси" |  |
| Изграждане на модул "Управление на документи" | * Генериране на документи по готови шаблони; * Общи документни шаблони; * Автоматично задаване на уникален входящ номер; * Автоматично попълване на наличната информация при попълване на документи; * Разпределяне на документите по отдели и служители; * Електронни уведомления за нови и просрочени документи и събития, възможност за ескалиране на събития при просрочване; * “История” на документите. |
| Изграждане на модул "Интеграция" | * Механизъм за конвертиране към вътрешен шаблон; * Каталогизиране на документите. |
| Изграждане на модул "Управление на задачи" | * Създаване и редактиране на задачи с различни атрибути: начало, срок, приоритет, статус; * Разпределяне на потребителите по задачите; * Следене на статуса по изпълнение на задачите; * Ескалиране на задачи. |
| Изграждане на модул "Изискване на достъп до документ" |  |
| Изграждане на модул "Справки" |  |

Разделянето се прави за по-лесно боравене с функционалностите. Детайли могат да бъдат намерени в UCMS-Software Requirements Specification.doc. В итерациите за разработване списъкът горе ще се приоритетизира и спрямо него ще се определят функционалностите за разработка за всяка итерация.

В следващата таблица са посочени фазите и основните дейности по тях (за повече информация UCMS-Project Schedule.mpp)

|  |  |  |  |
| --- | --- | --- | --- |
| **Фаза** | **Итерация** | **Primary Objective (risks/use cases addressed)** | **Планирано начало/край** |
| Inception  /Планиране/ | I1 | * Формиране на екипите; * Задаване на практическите задачи; * Дефиниране на роли и отговорности; * Дефиниране на потребителските изисквания; * Разработване на главен план за изпълнение на проекта; * Разработване визия на проекта; * План за управление на качеството; * Разработване на начална версия на речник; * Конвенция за писане на код; * Разработване на модел на инфраструктурата; * Разработване на списък с рискове; * Завършване на визията на проекта; | 28.09.2012 – 06.10.2013 |
| Elaboration  /Детайлизиране/ | E1 | * Ревизия на главния план на проекта * Допълване на речника; * Ревизия на списък с рискове; * Ревизия на плана за управление на качеството; * Ревизиране на визията на проекта; * Ревизия на модела на инфраструктурата; * Разработване на спецификация на изискванията; * Разработване на модел на данните; * Започване на разбота по документа с потребителските случаи. | 06.10.2013 – 03.11.2013 |
|  | Е2 | * Завършване на документа с потребителските случаи; * Завършване на софтуерната архитектура; * Създаване на план за следваща итерация; * Завършване на архитектурен прототип; * Тестване на архитектурен прототип; * Актуализиране на списъка с рисковете. |  |
| Construction  /Изграждане/ | C1 | * Ревизиране на документи;   Разработване на функционална група едно;   * Тестване на функционална група едно; | 04.11.2013 – 22.11.2013 |
|  | C2 | * Разработване функционална група две; * Тестване на функционална група две; * Разработване на план за внедряване и обучение; * Разработване на материали за инсталиране, администриране, поддръжка; * Разработване на материали за обучение. | 04.11.2013 – 21.12.2013 |
| Transition  /Предаване/ | T1 | * Мигриране на данни * Функционални тестове * Курсове за обучение | 23.12.2013 – 17.01.2014 |

Преди всяка итерация ще се изготвя детайлен план.

## **Детайлен план за итерация Е1**

Виж UCMS-Iteration-Plan-E1.

## **Детайлен план за итерация** C1

Виж UCMS-Iteration-Plan-C1.

## **Детайлен план за итерация** C2

Виж UCMS-Iteration-Plan-C2.

1. **Човешки ресурси**

Екипът е съставен от седем човека, които в зависимост от фазите на проекта ще изпълняват RUP ролите описани в профилите им. При нужда, поради недостатъчно време или външни фактори може по изключение член на екипа да заеме RUP роля, различна от описаната.

1. **Отчетни резултати**

## Итерации и отчетни резултати

На таблицата по-долу е представен списък на отчетните резултати, които предвиждаме да предаваме по проекта и какво е очакваното им състояние в края на всяка итерация. Всеки от предаваните документи има отговорник

Бележки към таблицата:

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

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

Следва таблицата с отчетни резултати:



Всеки отговорник на документ е подпомаган от хора от екипа, имащи по-малко като количество работа през съответния период. На таблицата долу може да бъде видяно пълното разпределение на отговорници на документи и помощници.



|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

## Описание на отчетните резултати

*По-долу е представено описание на отчетните резултати.*

|  |  |  |
| --- | --- | --- |
|  | **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. |
|  |  |  |
| 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 изисквания | Спецификацията на изисквания отразява тези изисквания към системата.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: Изисквания по отношение на нефункционалните характеристики биват документирани и съгласувани.  Фаза Изграждане: Спецификацията на 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. |
|  |  |  |
| 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 | 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 и обучение | Планът за внедряване описва набора от задачи по инсталиране и тестване на разработения продукт с оглед успешното му предаване на потребителя. Включва анализ на потребностите от обучение и стратегия за провеждане на обучението.  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. |
|  |  |  |
| 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. |

1. **Бюджет**

Проектът се реазлизира като част от дисциплината "Практикум 3" от специалността "Софтуерно инженерство". Екипът за разработка (поръчителя) не получава никакво заплащане.

## **Бюджет за човешки труд**

Ще бъде ангажиран единствено системен администратор от университета за:

* Инсталиране, конфигуриране и поддържане на системата Jira;
* Инсталиране на Nexus, хранилище за артефакти;
* Инсталиране, конфигуриране и поддръжка на SVN;
* Инсталиране, конфигуриране и поддръжка на виртуална машина за разработване и тестване на сървърите на университета, тъй като «Екип 2» не разполага със собствени ресурси за изграждане среда за разработка;
* Нужните консумативи се поемат от «Екип 2».

## **Бюджет за софтуер и консумативи**

Подходът за разработка 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.

* Продуктите от Microsoft Office са взети с помощта на MSDN инициативата към РУ;
* За език за разработка е избран java и Eclipse средата, които са/могат да бъдат свалени безплатно. При нужда от готови решения ще се търси софтуер с отворен код.