Нaционална информационна система за Електронно здраве (E-Health)

План за разработка на софтуер

Версия 1.0

История на промените

|  |  |  |  |
| --- | --- | --- | --- |
| **Дата** | **Версия** | **Описание** | **Автор** |
| 2014/10/11 | 1.0 | Създаване на документа | Малвина Макариева |

Съдържание

1. Въведение 4

1.1 Цел 4

1.2 Обхват 4

1.3 Дефиниции, акроним и абревиатури 4

1.4 Препрадки 4

2. Обзор на проекта 5

2.1 Цели на проекта, обхват и обекти 5

2.2 Ограничения 5

2.3 Работа по проекта - продукти 5

2.4 Развитие на софтуерния план 6

3. Организация на проекта 6

3.1 Организационна структура 6

3.2 Външни групи 6

3.3 Роли и отговорнисти 6

3.3.1 Прфили на членовете на екипа 9

4. Управление на процеса 10

4.1 Оценка на проекта 10

4.2 План на проекта 10

4.2.1 Фази на плана 10

4.2.2 Итеративна разработка 17

4.2.3 Версия за предаване 18

4.2.4 План график 18

4.2.5 Ресурси използвани за осъществяването на проекта 18

4.2.6 Бюджет 18

4.3 Планове на итерациите 19

26.1 Контрол на проекта 20

26.1.1 План за контрол на „План за разработка на софтуер“ и „Графика на проекта“ 20

26.1.2 План за контрол на бюджета 20

26.1.3 Контрол на чачеството 21

26.1.4 План на докладите 21

26.1.5 План за измерване 21

26.2 План за управление на риска 21

26.3 Външни зависимости. 21

27. Технически план за изпълнение 21

27.1 Задачи за разработка 21

27.2 Методи, инструменти и техника за разработка 21

27.3 Инфраструктурен план 21

27.4 План за приемане на продукта 21

28. Подпомагане на процеса по планиране 22

28.1 Процедура по искане на промяна 22

28.2 План за оценка 22

28.3 План за документацията 22

28.4 План за управление на качеството 22

28.5 Разрешаване на проблеми и последващи действия 22

План за разработка на софтуер

# Въведение

Главният план за разработка на софтуерен проект е предназначен за изпълнителят 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.

## Цел

Цел на настоящия софтуерен план е разпределение на проекта по итераци и вместването на цялостната разработка в срока от 15 седмици.

## Обхват

Проекта обхваща планирането, детайлизирането, изграждането и предаването на проекта.

## Дефиниции, акроним и абревиатури

Всички дефиниции, акроними и абревиатури могат да бъдат намерени във файла „ Речник“. Използваните в този документ в табицата.

|  |  |  |
| --- | --- | --- |
| Съкращение | Пълно наименование | Описание |
| E-Health | E-Health | Английско наименование на Национална информационна система за електронно здраве |
| НСЕЗ | Национална система за електронно здраве | Национална система за електронно здраве |
| НЗОК | Национална здравноосигурителна каса | Управляващ орган към здравното министерство на България, което разпределя средствата свързани със здравноосигурените лица. |
| БЛС | Български лекарски съюз | Национална структура, в която членуват всички пратикуващи лекари в България |
| БЗС | Български зъболекарски съюз | Национална структура, в която членуват всички пратикуващи стоматолози в България |
| Е-рецепта | Електронна пецепта | Проект на НЗОК |
| RUP | Rational Unified Process | Методология за разработка на софтуерни проекти |
| SCRUM | SCRUM | Итеративна, инкрементална рамка за управление на проекти |
| SCRUM BUT | SCRUM BUT | Методологията SCRUM, но не изпълнявана напълно по правилата ѝ. |
| Jira | Jira | Софтуер за следене на задачи. ­­­­­­­­­­­­­­­­­­­­­­­­­­­­­ |

## Препрадки

* "Списък с рисковете";
* "Визия";
* "План за управление на качеството".

# Обзор на проекта

## Цели на проекта, обхват и обекти

Основни цели на проекта са проектирането и програмното разработване на Национална информационна система за електронно здраве "E-Health". Системата ще служи на пациенти, лекари и стоматолози.. В системата ще се съхранява информация за направените прегледи, предписаните лекарства и закупените от пациента такива.

## Ограничения

Ограноченоята към този проект са няколко:

* Ограничено време – времето за разработка е обвързано с продължителността на един семестър в Русенския университет „Ангел Кънчев“, а именно 15 седмици за целия процес по проектиране и програмиране;
* Екип – екипа е съставен от студенти. Гояма част от тях нямат опит в писането и разработката на подобен голям проект. Друго ограночение свързано с екипа е заетоста на членовете му. Около 55% от членовете на екипа работят на пълен работен ден. Екипът е малък, но с желание за развитие и научаване на нови технологии в процеса на разработката.

## Работа по проекта - продукти

| Наименование | Дата на предаване | Изпълнител |
| --- | --- | --- |
| План за разработка на софтуерен продукт | 25.10.2014 г. | Малвина Макариева |
| План за управление на качеството | 25.10.2014 г. | Лиляна Маринова |
| Списък с рисковете | 25.10.2014 г. | Малвина Макариева |
| Речник | 8.11.2014 г. | Малвина Макариева |
| Визия | 25.10.2014 г. | Михаил Радков |
| Бизнес модел | 25.10.2014 г. | Светослав Николов |
| Модел на потребителските случаи | 08.11.2014 г. | Симеон Илиев |
| Спецификация на софтуерните изисквания | 25.10.2014 г | Лиляна Маринова |
| Конвенция за писане на код | 25.10.2014 г. | Михаил Радков |
| Инструменти | 25.10.2014 г. | Михаил Радков |
| Модел на данните | 08.11.2014 г. | Светослав Николов |
| Модел на инфраструктурата | 25.10.2014 г | Михаил Радков |
| Дизайн модел | 08.11.2014 г. | Симеон Илиев |
| Софтуерна архитектура | 08.11.2014 г. | Росен Мартев |
| Модел на имплементация | 24.01.2015 г. | Росен Мартев |
| Прототип | 08.11.2014 г., 23.11.2014 г., 2.12.2014 г. | Разработчици |
| Завършена система | 17.01.2015 г. | Разработчици |
| Главен план за тесване | 08.11.2014 г. | Михаил Великов |
| Тестов модел | 23.11.2014 г. | Михаил Великов |
| Резултати от тестването | 23.11.2014 г., 2.12.2014 г., 17.01.2015 г., 24.01.2015 г. | Михаил Великов |
| План за внедряване | 23.11.2014 г. | Михаил Радков |
| Материали за инсталиране, администриране и поддръжка | 23.11.2014 г., 2.12.2014 г. 17.01.2015 г., 24.01.2015 г. | Михаил Радков |
| План за обучение | 23.11.2014 г. | Малвина Макариева |
| Материали за обучение | 23.11.2014 г. | Малвина Макариева |

таблица1 Списък с предоставените продукти по дата и изпълнител

## Развитие на софтуерния план

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

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

фигура1 Организационна структура

## Външни групи

* Възложител – връзката ще се осъществява между ръководителя на проекта /Малвина Макариева/ и ръководителя на проекта от страна на възможителят – Иван Станев;
* Външни системи, с които желаем да свържем системата E-Health. Отговорно лице ще бъде Росен Мартев. Такива системи са системата на Националната здравноосигурителна каса /НЗОК/, Български лекарски съюз /БЛС/, Български заболекарски съюз /БЗС/, Електронна рецепта /Е-рецепта/.

## Роли и отговорнисти

Практикум 3 се води по методологията на RUP (Rational Unified Process), затова за всеки човек от екипа се дава определен профил. Той съдържа в себе си две или повече роли от RUP.

| 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. Софтуерният архитект определя главната структура, като съвкупност от различни погледни над системат, 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. |
| Отговорник по изискванията | 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 екипа 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. |
| Отговорник по качеството | Quality Manager | 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. |
| 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. |
| Анализатор на бизнес процеси | Business Process Analyst | Отговаря за изучаването на процесите в организацията, анализирането им и прилагането им в процеса на разработка. |
| Инженер по данните | Data Engineer | Отговаря за определянето на видовете данни и тяхната структура, които ще се използват в системата. |
| Дизайн мениджър | Design Manager | Определя част от системата по изискванията и архитектурата. |
| Системен анализатор | System Analyst | Управлява процеса по избиране и моделиране на потребителски случаи, избирайки ключовите за системата функционалности. |
| Системен администратор | System Administrator | Отговорен за целия хардуер, софтуер и инфраструктура, необходими при разработването на продукта. Прави инсталации, конфигурации и резервни копия. |
| 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, инструменти 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. |

таблица2 Роли и отговорности

Други изпълнявани роли в екипа са според SCRUM

|  |  |  |
| --- | --- | --- |
| Българско наименование | Английско наименование | Отговорност |
| Продуктовият собственик | Product Owner | Product Owner отговаря за максимизирането на полезността на продукта и на работата на Екипа от Разработчици. Начинът, по който това се постига, може да варира в широки граници при различните организации, Scrum Екипи и хора. |
| Ръководител на SCRUM екипа | Scrum Master | Scrum Master отговаря за това Scrum да бъде правилно разбран и следван. Scrum Master постига това, осигурявайки Scrum Екипът да се придържа към теорията, практиката и правилата на Scrum. Scrum Master е лидер-слуга на Scrum Екипа. Той служи на Екипа от разработчици и на организацията. |
| Екип от разработчици | Scrum Team | Екипът от рРазработчици се състои от професионалисти, чиято работа е да произвеждат потенциално използваем Инкремент от "Готов" продукт в края на всеки Спринт. Само членове на Екипа от разработчици участват в самото създаване на Инкремента.  Екипите от Разработчици имат следните характеристики:   * Те са самоорганизиращи се. Никой (дори Scrum Master) не може да казва на Екипа от Разработчици как точно да превръща Product Backlog в Инкременти от годна за потенциално използване функционалност; * Екипите от Разработчици са многофункционални, притежаващи като екип всички умения, необходими за произвеждането на Инкремент от продукта; * Scrum не признава други титли или позиции за членовете на Екипа от Разработчици освен Разработчик (Developer), независимо каква работа извършва даден човек. Изключения от това правило няма. * Scrum не признава под-екипи в рамките на един Екип от Разработчици, независимо от различните области, които трябва да бъдат адресирани – като например тестване или бизнес анализ. Няма изключения от това правило * Отделни членове на Екипа от Разработчици може да имат специализирани умения и области, върху които са фокусирани, но отговорността принадлежи на Екипа от азработчици като цяло. |

таблица3 Роли по SCRUM

Важно уточнение е, че методологията на SCRUM не се прилага напълно дословно, в следствие на което екипа работи по SCRUM BUT. Това е разбиране на методологията и прилагането ú в съответствие с екипа и задачите, а не дословното ú изпълнение.

### Прфили на членовете на екипа

Профилите на членовете на екипа както следва:

#### Профил: „Ръководител на екипа“

Ролята се изпълнява от Малвина Макариева. Тя се занимава с плана за разработка, графиците, качеството, План за обучение и Ръководство на потребителите:

* Ръководител на проекта (Project Manager);
* Технически документатор (Tech Writer);
* Отговорник по качеството (Quality Manager) – тази роля се изпълнява само на половина.

#### Профил: „Бизнес аналитик и ижинер по качеството“

Изпълнява се от Светослав Николов и Лиляна Маринова. В екипа по качеството са включени Лиляна Маринова и Малвина Макариева. Те отговарят за качеството на разработката. Светослав Николов изпълява ролята на Бизнес аналитик с помоща на екипа по качеството.

* Анализатор на бизнес процеси (Business Process Analyst);
* Системен анализатор (System Analyst);
* Събирач на изисквания (Requirements Specifier).

#### Профил: „Бизнес аналитик и инженер по данните“

Изпълнява се от Светослав Николов. Той ще отговаря за бизнес анализа и модела на данните за системата.

* Анализатор на бизнес процеси (Business Process Analyst);
* Инженер по данните (Data Engineer).

#### Профил: „Разработчик“ и „Софтуерен архитект"

Изпълнява се от Росен Мартев, Михаил Радков и Симеон Илиев. Разработчика отговаря за разработката и свързаните с нея документи.

* Разработчик(Implementer);
* Софтуерен архитект (Software Architect);
* Интегратор (Integrator).

#### Профил: „Разработчик“ и „Системен администратор“

Отговорник Михаил Радков. Разработва системата, като системен администратор се занимава с изграждане на средата за разработка, средата за тестване, инструментите, които ще се използват в процеса на разработка.

* Разработчик(Implementer);
* Инструментален специалист (Tools specialist).
* Интегратор (Integrator)

#### Профил: „Разработчик и Дизайн мениджър“

Изпълнява се от Симеон Илиев. Той отговаря за изработката на дизайн модела на системата и участва пряко в разработката ѝ.

* Разработчик(Implementer);
* Дизайн мениджър (Design Manager).

#### Профил: „Тест мениджър и тестер“

Изпълнява се от Михаил Великов.

* Тест мениджър (Test Manager);
* Тест дизайнер (Test Designer);
* Тестер (Tester).

# Управление на процеса

## Оценка на проекта

Критериите за оценка за 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.

## План на проекта

### Фази на плана

#### Списък с продуктите за предаване и тяхното описание

|  | **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 изискванията\* | 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\* | Софтуерната архитектура представя комплексен архитектурен изглед нa системата, използвайки за целта редица различни архитектурни разрези, показващи отделни нейни аспекти. |
| 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. |

таблица4 Описание на документите за разработката

#### Диграми на екипа свързани с разпределенето на работата по целия проект.

#### Междинни точки

RUP определя четири фази при разработката на софтуерни проекти. Във всяка фаза екипът по изпълнението се фокусира над конкретен аспект от проекта и свързаните с него редица междинни точки (milestones). Тези междинни точки помагат на ръководителя на проекта да направи оценка за напредъка на проекта 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 по време на тази фаза.

### Итеративна разработка

По-голямата част от проектите по RUP, са по дефиниция итеративни. RUP е постепенен процес, при който цялостният проект е разделен на фази и повторения. Итерациите са риск задвижвани- това е, ориентирана към намаляване на рисковете. При всяка итерация изпълнителя трябва да достави изпълнимия софтуер, който е доказуем и проверим срещу изискванията на проекта и потребителските случаи.

Ръководителят на проекта използва детайлните планове на итерации за упавление на проекта. 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 ѝ;
* 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. Мениджърът съчетава итеративен ориентиран към рисковете подход с прецизирани задачи за точно определяне на плана на итерацията.

### Версия за предаване

| **Врсия** | **Модул** | **Финкционалност/под модул** |
| --- | --- | --- |
| Прототип – 1.0  (23.11.2014 г.) | Електронен картон | Обща информация |
| Електронен картон | Алергии |
| Електронен картон | История на прегледите |
| Електронен картон | Имплантанти |
| Електронен картон | История на влизанията в профила |
| Прототип – 1.1  (21.12.2014 г.) | Електронен картон | История на посещенията |
| Електронен картон | Регистрация на ОПЛ |
| Електронен картон | Лекарства |
| Електронен картон | Диспансерни регистации |
| Електронен картон | Хоспиталицации |
| Административен модул | Административна час |
| Административен модул | Функционалности на оператора |
| Регистри | Лекари |
| Регистри | Стоматолози |
| Прототип – 1.2  (17.01.2014 г.) | Електронен картон | Лабораторни изследвания |
| Електронен картон | Медицински процедуръ |
| Електронен картон | Елекронен зъбен картон |
| Помощ | Легенда |
| Помощ | Често задавани въпроси |
| Помощ | Видео уроци |

таблица 5 Списък с версиите за предаване

### План график

### Ресурси използвани за осъществяването на проекта

#### Продукти използвани за разработката

Списък с продуктите и инструментите за разработката на проекта може да бъде намерен в документа „Инструменти“.

#### Човешки ресурс

Екипът е съставен от седем човека, които в зависимост от фазите на проекта ще изпълняват различни роли по методологията RUP. Ролите са описани е профилите им. В случай, че някой от екипа не успява да се справя със задачите си за необходимото време друг по-малко зает член на екипа ще му указва помощ. Ако член на екипа не изпълнява съвестно задълженията си и отказва да работи по проекта, то той може да се изключи от екипа.

#### План за обучение

##### План за обучение на екипа

* + - Екипа ще бъде запознат с начините за използване на Jira, като предварително ще бъде определен ден за това и обучението ще се проведе от Михаил Радков на целия екип;
    - План за обучение на екипа по разработката за работа с изградената среда - ще се преведе след изграждане на средата от Михайл Радкков.

##### План за обучение на клиента /пациенти и медицински лица.

* + - След изграждане на системата и написване на „Ръковдството на потребителя“, то ще бъде достъпно в интернет;
    - След пускането на системата ще се изгради допълнителен сайт с уроци за работа на различните потребители – пациенти, медицински лица и фармацевти. Всеки от потребителите ще може да избере удобно време, за да мине курса на обучение самостоятелно.

### Бюджет

Бюджет за проекта не е определен поради образователният характер на задачата. Екипа няма да получава заплащане за извършените услуги.

Всички продукти за включени в процеса на разработка на E-Health са безплатни. Списък с използваните инструменти можете да намерите в документа „Инстументи“.

## Планове на итерациите

Таблица със итерациите и документите за предаване на всяка итерация.

|  | **Наименование** | **I1** | **E1** | **E2** | **C1** | **C2** | **C3** | **T1** | **Изпълнител** | **Роля** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | План за разработка на софтуерния продукт | **s** | **r** | **s** | **r** | **r** | **r** | **c** | Малвина Макариева | PM |
| 2 | План за управление на качеството | **c** |  |  |  |  |  |  | Лиляна Маринова | QA |
| 3 | Списък на рисковете | **s** | **r** | **r** | **r** | **r** | **r** | **c** | Малвина Макариева | PM |
| 4 | Детайлен план за итерация | **c** | **c** | **c** | **c** | **c** | **c** |  | Малвина Макариева | PM |
| 5 | Речник | **e** | **e** | **s** | **r** | **r** | **r** | **c** | Малвина Макариева | TW |
| 6 | Визия | **c** |  |  |  |  |  |  | Михаил Радков |  |
| 7 | Бизнес модел\* | **e** | **s** | **r** | **r** | **r** | **r** | **c** | Светослав Николов | BA |
| 8 | Модел на потребителските случаи\* | **e** | **e** | **s** | **r** | **r** | **r** | **c** | Симеон Илиев | BA |
| 9 | Спецификация на изискванията\* | **e** | **s** | **r** | **r** | **r** | **c** |  | Лиляна Маринова | Req.E |
| 10 | Конвенции за писане на код | **c** |  |  |  |  |  |  | Михаил Радков | Sys.Adm. |
| 11 | Инструменти | **e** | **s** | **s** | **r** | **r** | **r** | **c** | Михаил Радков | Sys.Adm. |
| 12 | Модел на данните\* | **e** | **e** | **s** | **r** | **r** | **r** | **c** | Светослав Николов | Dat.E. |
| 13 | Модел на инфраструктурата\* | **e** | **s** | **r** | **r** | **r** | **r** | **c** | Михаил Радков | Sys.Adm. |
| 14 | Дизайн модел\* |  |  | **s** | **r** | **r** | **r** | **c** | Симеон Илиев | Design.M |
| 15 | Софтуерна архитектура\* |  |  | **s** | **r** | **r** | **r** | **c** | Росен Мартев | Soft.Arch. |
| 16 | Модел на имплементацията\* |  |  | **e** | **e** | **e** | **s** | **c** | Росен Мартев | Dev. |
| 17 | Прототипи |  |  |  | **c** | **c** |  |  | Росен Мартев, Михаил Радков, Симеон Илиев | Dev. |
| 18 | Завършена система |  |  |  |  |  | **s** | **c** | Росен Мартев, Михаил Радков, Симеон Илиев | Dev. |
| 19 | Главен план за тестване |  | **e** | **s** | **r** | **r** | **r** | **c** | Михаил Великов | TM |
| 20 | Тестов модел\* |  | **e** | **e** | **s** | **r** | **r** | **c** | Михаил Великов | TE |
| 21 | Резултати от тестовете\* |  |  |  | **c** | **c** | **c** | **c** | Михаил Великов | ТЕ |
| 22 | План за внедряване |  |  |  | **s** | **s** | **s** | **c** | Михаил Радков | SYs.Adm. |
| 23 | Материали за инсталиране, администриране, поддръжка\* |  |  |  | **s** | **s** | **s** | **c** | Михаил Радков | Sys.Adm. |
| 24 | План за обучение |  |  |  | **s** | **r** | **r** | **c** | Малвина Макариева | PM |
| 25 | Материали за обучение |  |  | **e** | **s** | **s** | **s** | **c** | Малвина Макариева | TW |

таблица6 Списък с документите и развитието им по итерации

|  |  |  |  |
| --- | --- | --- | --- |
| **Резултати** | | **Фази** | |
| **Код** | **Описание** | **Код** | **Описание** |
| e (evolved) | В процес на разработка (по резултата се работи) | I (inception) | Планиране |
| s (stable) | Приет (утвърден) | E (elaboration) | Детайлизиране |
| r (revised) | Ревизиран (актуализиран след приемане, при необходимост) | C (construction) | Изграждане |
| c (completed) | Завършен (замразен) | T (transition) | Предаване |

таблица7 Легенда

## Контрол на проекта

### План за контрол на „План за разработка на софтуер“ и „Графика на 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 проблемите по проекта се отстраняват своевременно. Задачите за всяка итерация се определят преди началото ѝ, като задачата на всеки член от екипа е да направи приблизителна оценка на поставените му задачи. По този начин „Екип Едно” ще може по-лесно да оцени усилието, нужно при разработка;
* Провеждане на седмични онлайн срещи и изготвяне на отчети. Всяка седмица, вторник и четвъртък от 20:00 екипът провежда срещи, както и в събота в Русенки университет „А. Кънчев“, по изпълнението на проекта. Анализират се изпълняваните в момента задачи и прогреса по тях. Срещите се провеждат по SCRUM BUT 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 Едно и поръчителите от Русенки университет „А. Кънчев“. 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.

### План за контрол на бюджета

Такъв план няма да се разработва поради лиспсата на определен такъв в следствие от обучителния характер на проекта.

### Контрол на чачеството

Описано е в документа „План за управление на качеството“.

### План на докладите

План за докладите е описан в „Плана за управление на качеството“

### План за измерване

Метрики за измерване са описани в „План за управление на качеството“

## План за управление на риска

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 „Списък с рисковете“.

## Външни зависимости.

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

Конкретният „Екип Едно” ще работи по E-Health, 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.

# Технически план за изпълнение

## Задачи за разработка

Задачите за разработката могат да се видят в Jira, както и в „Графика на плана“.

## Методи, инструменти и техника за разработка

Тази изформация е достъпна във всички документи, които са гайдове за изработката на друг документ.

## Инфраструктурен план

На местата, на които е необходимо определянето на инфраструктурен план ще се използват различни стандарти, като ISO, които да гарантират качеството.

## План за приемане на продукта

Графика за приемане на продукта може да се види в Графика на проекта.

# Подпомагане на процеса по планиране

## 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:

## План за оценка

Описан е в точка 4.1

## План за документацията

Плана за документацията може да бъде видян по-горе в документа или качен в Google Drive с име EH-DeliverablesList-v1-0.xlsx.

## План за управление на качеството

Този план е описан в отделен документ с наименованието „План за управление на качесвото“.

## Разрешаване на проблеми и последващи действия

Информацията може да бъде намерена в документа „ План за управление на качеството".