"

План за управление на качеството

Версия 1.1

История на промените

|  |  |  |  |
| --- | --- | --- | --- |
| **Дата** | **Версия** | **Описание** | **Автор** |
| 2014/10/08 | 1.0 | Създаване на документа | Лиляна Маринова |
| 2014/10/11 | 1.1 | Допълване на документа | Лиляна Маринова |

Съдържание

1. Въведение 4

1.1 Предназначение 4

1.2 Обхват 4

1.3 Дефиниции, съкращения, абревиатури 4

1.4 Препратки 4

1.5 Обзор 4

2. Цели на качеството 4

3. Управление 4

3.1 Организация 4

3.2 Задачи и отговорности 5

4. Документация 5

5. Стандарти и указания 6

6. Метрики 6

7. План на прегледи и одити 7

7.1 Предаване и приемане на итерация 7

8. Оценки и тестове 7

9. Инструменти, техники и методологии 7

10. Записи по качеството 7

11. Управление на риска 7

План за осигуряване на качеството

# Въведение

## Предназначение

Предназначението на този документ е да се опишат задачите и отговорностите, които трябва да се извършат от екипа, отговорен за осигуряване на качеството.

## Обхват

Документът обхваща всички фази на проектиране на продукта, където има зададени стандарти за качеството. На всеки етап от разработката на софтуерния продукт ще се правят проверки за качеството.

## Дефиниции, съкращения, абревиатури

Дефинициите, съкращенията и абревиатурите са описани в документа „Речник”;

## Препратки

В този раздел са представени всички документи, към които има препратки от ПУК (План за управление на качеството) на този етап от разработката на проекта:

* „Списък на рисковете”;
* „План за разработка на софтуерния продукт”;
* „Спецификации на софтуерните изисквания”.

## Обзор

До края на този документ са описани:

* показателите за качеството;
* управление от гледна точка на организацията;
* необходимата документация за проекта;
* задачите и отговорностите;
* метриките за измерване на софтуерния продукт и изчисляване на отработеното време;
* предаване на итерациите;
* стандарти и указания;
* инструменти.

# Цели на качеството

Показателите, които засягат качеството на разработвания продукт са описание в документа „Спецификации на софтуерните изисквания“.

# Управление

## Организация

Роли, отговорни за осигуряване на качеството:

* Ръководител на проекта (Project Manager) - разпределя отговорностите в екипа, определя приоритетите и координира комуникацията с клиента и потребителите. Той е отговорен за подсигуряване на целостта на качеството на проекта;
* Функционален тестер (Unit Tester) - 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;
* Тест мениджър (Test Manager) - отговорен за целия процес по тестване и качество на тестовете;
* Отговорник по качеството(Quality Manager) – описва задачите и отговорностите, които ще се извършват от екипа, за да се осигури качеството на софтуерния продукт.

RUP (Rational Unified Process) препоръчва оценката на продукта да се направи от независим тестов екип. Поради факта, че проектът се разработва в рамките на учебния процес и екипът е малък, за целта за осигуряване на качеството ще работи екип QA (Quality Assurance) - Лиляна Маринова и Малвина Макариева, като при необходимост ще получават помощ от другите членове на екипа. Екип QA ще прави проверки по време на всяка фаза от проекта с цел при откриване на неточности и несъответствия те да бъдат решавани в кратки срокове и да се поддържа определен стандарт на качество.

## Задачи и отговорности

Този раздел описва различните задачи за осигуряване на качеството. Описани са и ролите, които отговарят за изпълнението на задачите.

* Външни одити – реализира се от един или повече одитори, които не са членове от екипа по разработка. Те реализират независима проверка на софтуерния продукт. Ще се извършват на всяка итерация;
* Вътрешни одити – членовете от екипа на разработка ще осъществяват преди всяка итерация проверки на разработените до момента документи, като ще следят дали има несъответствия и проблеми. Ролите, отговорни за провеждането на вътрешните одити са екипът QA;
* Интервюта за извършените тестове – ще се реализират от QA, които ще интервюират тестерите. Те ще ни носят информация за направените тестови случаи. Ще се извършват при всеки извършен тест;
* Интервюта за извършената работа - ще се реализират от QA, които ще интервюират всички членове на екипа за свършената работа. Ще се извършват по време на всяка итерация.

# Документация

Този раздел включва документите, които ще отговарят за това качеството на продукта да отговаря на зададените му изисквания:

* План за разработка на проекта – създава се във фаза Планиране;
* План за тестване – създава се във фаза Детайлизиране;
* Спецификации на софтуерните изисквания – определят се функционалните и нефункционалните изисквания във фаза Планиране;
* Софтуерна архитектура – създава се във фаза Планиране;
* Документация за потребителя – създава се в последната фаза на Изграждане.
* Бизнес модел – създава се във фаза Планиране;
* Модел на потребителските случаи – създава се във фаза Планиране;
* Модел на данните – създава се във фаза Планиране;
* Тестов модел – създава се във фаза Детайлизиране;
* Прототип – създава се в последната фаза на Детайлизиране.

# Стандарти и указания

В този раздел са упоменати всички стандарти и насоки, които ще се използват по време на разработката на проекта:

* Java Code Conventions – конвенцията на Oracle за писане на програмен код;
* UML 2.0 – език за графично моделиране на софтуер;
* BPMN 2.0– представяне на бизнес процесите;
* XML – стандарт, който определя правила за създаване на специализирани езици и синтаксисът, на който тези езици трябва да се подчиняват;
* AJAX – похват е разработката на уеб базирани приложения, който ще използваме за нашата система.

Указания за именуване на документите:

1. В началото е номера на документа, след това са инициалите на системата (EH), след което се поставя тире и се изписва името на файла със английското му значение, като всяка дума започва с главна буква и не се поставя разстояние между думите. Рапортите се именуват по следния начин: в началото са инициалите на системата (EH), след което се поставя тире и се изписва Report, поставя се тире, имената на човекът попълнил рапорта, като не се оставя разстояние между тях и всяко от имената е с главна буква, поставя се тире и се попълва датата, като се пише първо годината, след това тире, пише се месецът, поставя се тире и се пише датата. Пример "2-EH-Report-LilqnaMarinova-2014-10-12";
2. Версия номер – коя версия на документът е.

Пример: 2-EH-QualityAssurancePlan-v1-0

Указания за писане на препратки:

* Препратките да се изписват по следния начин – изписва се името на документа в кавички

Пример: Препратка към следния документ:

* „План за управление на качеството”.

Указания за писане на дати в таблиците "История на промените":

* Датата се пише по следния начин - първо се изписва годината, след това се поставя "/", след нея се изписва месецът, отново се поставя "/" и се изписва датата. Пример: 2014/10/08.

# Метрики

Метрики за проследяване на грешки:

* Общ брой намерени грешки – общия брой намерени грешки в кода на системата. Ще бъдат обхванати всички файлове, които съдържат програмен код. Метриката ще се отчита като се вземат в предвид грешките въведени в JIRA на BitBucket, както и чрез таблици и интервюта, понеже филтрите в BitBucket са малко;
  + Неразрешени грешки – общият брой открити и неразрешени грешки в системата. Стойността ще е част от общия брой грешки;
  + Разрешени грешки – общият брой открити и разрешени грешки в системата. Стойността ще е част от общия брой грешки;
* Реализирани функционалности – общия брой реализирани функционалности и броя оставащи да се изпълнят;
* Програмирани Use Case(UC) – общия брой програмирани UC. Дава ни информация за това колко още UC ни остават да се програмират и дали сме в графика, предвиден за изпълнението на задачата. Информацията за програмираните UC ще се набавя от рапорти от програмистите;
* Общ брой тестови случаи - включва броя изпълнени тестови случаи(брой успешни и неуспешни) и броя оставащи да се извършат;

Приоритет на грешките(бъговете) - за отчитането на метриката ще се взима под предвид грешките въведени в уеб базираната система JIRA на BitBucket и тяхното ниво на сериозност: critical, major, trivial, minor и blocker. Като blocker и critical са от най – висок приоритет за разрешаване, със среден приоритет са major, с нисък приоритет са minor и trivial.

Метрики за определяне на качеството на процеса на разработка:

* Общото време отнело за разработката на проекта в човекочаса – включва времето за разработка на проекта на всяка от итерациите. Метриката ще се отчита на базата на свършената работа описана в JIRA на BitBucket, както и чрез таблици и интервюта, поради малкия брой филтри в BitBucket. В коментарите на задачите всеки от членовете на екипа ще записва времето необходимо за изпълнението на задачата;
  + Предвидено време в човекочаса – времето предвидено за проектирането на целия софтуерен продукт. Стойността ще е част от общото време отнело за разработката на проекта;
  + Изпълнено време в човекочаса – времето, което реално е отнело за разработката на целия софтуерен продукт. Стойността ще е част от общото време отнело за разработката на проекта;

## Процес

При добре развиващият се процес съотношението между отворени и затворени задачи и грешки трябва да е в полза на затворените такива. За да бъде определен процеса като добре развиващ се, затворените задачи и грешки трябва да са над 60% от общия брой такива. Пример: имаме общо 100 на брой грешки и задачи, и над 60 от тях са затворени, а останалите 40 са все още отворени. Това означава, че процесът се развива добре и членовете от екипа си вършат работата.

При зле развиващ се процес съотношението между отворени и затворени задачи грешки е в полза на отворените такива. За да бъде определен един процес като зле развиващ се, отворените грешки и задачи трябва да са над 60% от общия брой такива, като при отчитането на отворените следва да се провери, по кои се работи и по кои не, и с какъв приоритет са. Пример: имаме общо 100 грешки и над 60 от тях са отворени, а останалите 40 са затворени. Това означава, че процесът се развива зле и трябва да се наблегне на повече работа от страна на членовете на екипа.

Ако съотношението на отворените и затворените задачи и грешки е в полза на отворените такива, но по тях системно се извършва работа и се отразява след това, развитието на процеса може да бъде определено като поносимо. Или да остане неопределено за момента до повторното му отчитане.

# План на прегледи и одити

## Предаване и приемане на итерация

След всяка итерация се провежда среща между екипа по разработка и възложителите. На тези срещи екипа по разработката запознава възложителите с извършените по време на итерацията дейности и предава всички необходими артефакти за съответната итерация. След като възложителя е запознат с артефактите, той трябва да прецени дали те са с нужното качество. Ако възложителя прецени, че качеството на артефактите е задоволително, то той приема документите и итерацията се счита за предадена и подписана. Ако възложителя прецени, че качеството на артефактите е незадоволително, то възложителя връща документите за ревизия със съответните бележки по тях и се насрочва нова среща за предаване на итерацията. Тя се провежда във възможно най–кратки срокове.

Предаването и приемането на документи е описано в документа „Детайлен план за итерацията”.

# Оценки и тестове

Тестването на системата започва от момента, в който е разработен някой компонент на системата, който компонент може да бъде цял програмен модул, отделна функция или процедура. Всеки от разработчиците създават компонентни тестове върху кода, който са написали. Освен компонентно тестване ще се извършва и функционално тестване, което цели да провери дали разработваната система отговаря на софтуерните изисквания. Този вид тестване ще се осъществя, след като разработчиците са приключили с компонентите си тестове.

Оценката на тестване се определя от степента на успеваемостта му. Степените съответно са: незадоволяваща, задоволяваща и отлична. Критериите за определя на степента са: покритие на кода, резултатите от тестовете, проблеми при тестването, време за извършване на тестовете. На всеки от критериите се дават оценки като най - ниската е 1, а най - високата е 5. Оценяването се извършва от отговорника по качеството и екипа му. Тази информация се събира посредством репорти, попълнени от тестерите. За всеки тест се попълва отделен рапорт.

Останалите видове тестове ще бъдат разработени на следващите итерации и ще се опишат в документа "Главния план за тестване".

# Разрешаване на проблеми и последващи действия

При възникнал проблем, който не може да бъде решен вътре в екипа се описва в рапорт, който се попълват от членът на екипа, констатирал проблема. В рапорта се съдържа информация за:

* + кой член от екипа попълва рапорта;
  + описание на проблема;
  + степен на значимост на проблема - степените са: ниска, несъществена, умерена, съществена и висока.

След като е написан рапорта той се предава на ръководителя на проекта, който уведомява ръководството, за да се вземат необходимите мерки за разрешаването на проблема. Рапорта се предоставя и на отговорника по качеството, който попълва данните за проблема в таблица, като по този начин поддържа статистика за възникналите проблеми и статуса им(разрешени и неразрешени) както и времето отнело за разрешаването им. Таблицата представлява документ написан на Excel.

# Инструменти, техники и методологии

RUP(Rational Unified Process) – стандарт за разработване на големи софтуерни проекти.

JIRA – среда, съдържаща множество инструменти, които спомагат за управлението на разработката на софтуерния проект и проследяване на грешки.

ECLIPSE – среда за разработване на софтуер.

BitBucket - уеб - базирана услуга за хостинг на проекти, които използват Git система за ревизия.

# Записи по качеството

За целта ще се използва JIRA на BitBucket. Чрез нея ще се следи прогреса по зададените задачи на членовете на екипа, разработващ проекта. Ще се следи забавянето на задачите във времето посредством филтрите в BitBucket.

# Управление на риска

Всичко относно управлението на риска е упоменато в документа „Списък на рисковете”.