Система за електронна търговия Balkan Bay (BBay)

План за осигуряване на качеството

Версия 1.1

История на промените

|  |  |  |  |
| --- | --- | --- | --- |
| **Дата** | **Версия** | **Описание** | **Автор** |
| 2014.01.28 | 1.0 | Създаване на документа | Л. Маринова |
| 2014.01.31 | 1.0 | Ревизия, определяне на метрики и инструменти | Л. Маринова |
| 2014.02.01 | 1.0 | Ревизия, определяне на метрики и инструменти | Л. Маринова |
| 2014.02.03 | 1.1 | Ревизия, добавяне на още метрики | Л. 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 8

7.1 Предаване и приемане на итерация 8

8. Оценки и тестове 8

9. Инструменти, техники и методологии 8

10. Записи по качеството 8

11. Управление на риска 8

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 стандарт 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 следния документ:

* „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:

* Мениджър на проекта (Рroject Мanager) - 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;
* Интеграционен тестер (Integration Tester) - планира и изпълнява билдове на готова функционалност;
* 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 екипите са малки, 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, които трябва 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 oдити – 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 итерации - Е3 и Т1;
* 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 свършена работа– ще се реализират от екип QA. Те ще ни донесат информация за това дали членовете на екипа съвестно си вършат работата, дали си свършват задачите по график и спазват ли своите задължения. Ще се извършват между разработването на документите и предаването им на възложителя;
* Интервюта за извършените тестове – ще се реализират от QA, които ще интервюират Unit Tester. Те ще ни носят информация за направените тестови случаи. Ще се извършват преди предаването на итерации C1, C2, C3;
* Чеклисти – всеки член на екипа си води чеклисти със задачите, които има да свърши. Когато е готов с някоя от задачите си я отмята от чеклистата. Ще се извършват по време на разработката на продукта.

# 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;
* Детайлен план на итерациите – създава се във фаза Планиране и преди началото на всяка итерация;
* 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. Номер на документа – номера на документа по Deliverable List;
2. Номер на фаза – на коя итерация се предава документът;
3. Датата на която се предава гггг/мм/дд;
4. Името на файла със английското му значение;
5. Версия номер – коя версия на документът е;

Пример: 2-I1-140202-Quality Assurance Plan-v1-0.docx

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”.

Списъка с документите и техните пълни наименования се съдържат във файла – 00-Bill\_of\_materials.xlsx.

# 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;
* 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;
* Грешки по модул – броя открити грешки в системата по модули. Стойността ще е част от общия брой грешки;
* Реализирани функционалности – общия брой реализирани функционалности и броя оставащи да се изпълнят;
* Проектирани Use Case – общия брой проектирани Use Case и броя оставащи да се изпълнят. Дава ни информация за това колко още Use Case ни остават да се проектират и дали сме в графика, предвиден за изпълнението на задачата;
* Програмирани Use Case – общия брой програмирани Use Case. Дава ни информация за това колко още Use Case ни остават да се програмират и дали сме в графика, предвиден за изпълнението на задачата;
* Тествани Use Case – включва броя изтествани Use Case и броя оставащи да се тестват;
* Разрешени грешни Use Case – включва броя открити и разрешени Use Case;
* Неразрешени грешни Use Case – включва броя открити, но не разрешени Use Case;
* Размер на програмата– включва броя програмни редове, броя редове с коментари и броя празни редове.
* Метрика на Маккейб – чрез нея ще измерваме структурната сложност на програмата;
* Време за отстраняване на грешки(бъгове) – времето в човекочаса изразходено за отстраняването на грешките;
* Приоритет на грешките(бъговете) - за отчитането на метриката ще се взима под предвид грешките въведени в уеб базираната система JIRA 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;
* 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;
* Срок за изпълнение в човекочаса– времето, което е предвидено за изпълнение на задачите за всеки член на екипа. Метриката ще се отчита на базата на описаното време в JIRA;
* Срок за просрочване в човекочаса– реално просроченото време по изпълнението на задачите. Метриката ще се отчита на базата на описаното време в JIRA;
* Изпълнение на задачите – метриката отчита до колко съвестно членовете на екипа си изпълняват задачите и отговорностите. Метриката включва просрочено време за решаването на задачата, навременно ли е решаването на проблеми, ако има възникнали такива, присъствие на конферентните срещи.

# 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

Оценките и тестовете ще бъдат разработени на следващите итерации и ще се опишат в документите:

* Главния план за тестване;
* Резултати от тестването;

# 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.

QUICKR – инструмент за споделяне на информация между хората, участващи в разработката на проекта.

# Записи по качеството

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

# Управление на риска

Всичко относно управлението на риска е упоменато в документ:

* + „Списък на рисковете”;