Главен план за тестване

Версия 1.0

История на промените

|  |  |  |  |
| --- | --- | --- | --- |
| **Дата** | **Версия** | **Описание** | **Автор** |
| 2013.03.19 | 1.0 | Създаване на документа | Малвина Макариева |

Съдържание

1. Представяне 4

1.1 Предназначение 4

1.2 Обхват 4

1.3 Дефиниции, акроними и абревиатури 4

1.4 Връзки 4

1.5 Източници или мотиватори за тестове 4

2. Общ преглед на планираните тестове 4

3. Видове тестове 4

3.1 Unit Test /Компонентно тестване/ 4

3.2 Functional Test/Функционално тестване/ 5

3.3 User Interface Test /Тестване на потребителски интерфейс/ 5

3.4 Usability Test /Тестване за ползваемост/ 5

3.5 Security And Access Control Test /Тестване на сигурност и контрол на достъп/ 5

3.6 Regression Test /Регресивно тестване/ 5

3.7 Integration Test /Интеграционно тестване/ 6

4. Какво ще се тества 6

4.1 Преглед на планираните тестове 6

4.1.1 Преглед на функциите , които ще се тестват 6

4.1.2 Преглед на допълнителни функции, които ще се тестват 6

5. Подходи при тестване 6

5.1 Измерване степента на тестване 6

5.2 Идентифициране на тестовете 6

6. Критерий за вход и изход 6

6.1 Главен план за тесване 6

6.1.1 Входящ критерий на главния план за тестване 6

6.1.2 Изходящ критерий на главния план за тестване 6

6.1.3 Критерии за спиране и възобновяване 7

7. Продукти от тестването 7

8. Процес на тестване 7

8.1 Изисквания към средата 7

8.1.1 Базов хардуер на системата 7

8.2 Базови софтуериелементи на тестовата среда 7

8.3 Продуктивност и проддръжка 8

8.4 Отговорности при тестването 8

8.5 Обучение на тестери 9

8.6 Рискове за главния план за тестване 9

Главен план за тестване

# Представяне

## Предназначение

Предназначението на „Главния план за тестване“ е да опише:

* Този документ е описани на техниките, методите и подходите за тестване. Той е основен артефакт за организация и управление на процеса по тестване;
* Осигурява прозрачност за всички заинтересовани лица.

## Обхват

## Дефиниции, акроними и абревиатури

Информация за използваните дефиниции, акроними и абревиатури можете да намерите в специализирания документ „Речник“.

## Връзки

Документа кореспондира с „Тестов модел“ и „Резултати от тестване“

## Източници или мотиватори за тестове

* Функционални изисквания;
* Нефункционални изисквания;
* Use cases;
* Activity диаграми;
* Промени при изискванията;
* Елементи на дизайна;
* План за качество.

# Общ преглед на планираните тестове

* Unit Test;
* Functional Test;
* User Interface Test;
* Usability Test;
* Security And Access Control Test;
* Regression Test;
* Integration Test.

# Видове тестове

## Unit Test /Компонентно тестване/

Компонентното тестване е процес в тетсването, чрез който се тестват отделни единици (компоненти) от сорс код - един или повече програмни модула и техните контролни данни, процедури за използване, оперативни процедури с цел да се установи дали работят правилно. Под компонент се разбира най-малката част на едно програмно приложение, която може да бъде самостоятелно тествана. Компонент би могъл да бъде целия програмен модул, отделна функция, процедура, цял клас, или отделен метод. Компонентните тестове се създават от програмист или в някои случаи от white box тестери по време на процеса на разработка, за да се гарантира правилното поведение на софтуера и дали той покрива всички първоначално поставени изисквания.

## Functional Test /Функционално тестване/

Този вид тест ще се извърши въз основа на тестовия модел, който съдържа в себе си всички тестови сценарии за тестване на разработените функционалности.

Методът на функционално тестване се основава на тестване на дадена система според спецификацията на софтуерните компоненти, които биват тествани. Функционалностите се тестват като им се подават определени входни данни и се изследват изходните данни, за да се определи как работи дадената функционалност. Вътрешната структура на системата рядко се взема на предвид, но функционалното тестване определя как точно работи системата. Функционалното тестване обикновено включва няколко стъпки като:

* Определяне на функциите, които се очаква да бъдат изпълнявани от системата;
* Създаването на входните данни на базата на функционалните спецификации;
* Определянето на изходните данни на базата на функционалните спецификации;
* Изпълняването на самия тест;
* Сравняване на очакваните и действително получените данни.

## User Interface Test /Тестване на потребителски интерфейс/

Това е метод, при който потребителският интерфейс на системата се тества за грешки и доколко е достъпен. Този вид тестване може да проведе под различни форми, но най-често се съсредоточава върху определянето на това колко добре потребителите могат да взаимодействат със системата и откриването на грешки или проблеми по време на това тестване.

## Usability Test /Тестване за ползваемост/

Тестването за ползваемост е техника, която се използва в потребителски-ориентираните взаимодействия с цел създаването на оценка на системата, чрез прякото ú тестване от потребителите. Този вид тестване дава директна информация, за това как реалните потребители ще използват системата. Създаването на такъв вид тестове включва внимателното създаване на сценарии или реални ситуации, при които потребителят изпълнява списък или определена последователност от дадени действия, като през това време хора от екипа за тестване наблюдават как потребителя използва системата, за да извърши възложената му задача и си водят бележки, които да им послужат за подобряване на системата.

## Security And Access Control Test /Тестване на сигурност и контрол на достъп/

Този вид тестване е от изключителна важност, защото при този процес се определя дали системата защитава данните и дали изпълнява функционалностите според предназначенията и спецификациите им. Друг аспект на това тестване е да се определи точното ниво на достъп на определените потребители до данните в системата според правата, които имат.

Има шест основни концепции, които трябва да бъдат обхванати по време на тестването на сигурността:

* Поверителност (конфиденциалност) – мярка за сигурност, която защитава от разкриване на информация пред лица, различни от предназначения получател на тази информация.
* Интегритет – мярка, която цели да предостави възможност на получателят (потребителят) да определи дали предоставената му информация от системата е коректна.
* Удостоверяване (аутентикация) – процес, при който се удостоверява самоличността на потребителя най-често чрез въвеждането на име и парола или друг подобен метод.
* Упълномощаване (ауторизация) – метод, който определя дали даден потребител има право да извършва определение действия или операции в системата.
* Наличие – осигуряването на информация и услуги, които да бъдат готови за използване; информацията трябва да е налична за ауторизираните потребители.

Поради спецификата на проекта и ограниченията този вид тест няма да покрие всички основни концепции заложени в него.

## Regression Test /Регресивно тестване/

Регресивното тестване е вид софтуерно тестване, което се стреми да открие нови бъгове в системата и съществуващи функционални и нефункционални области, след направени промени в тях като подобрения, поправки на бъгове или промени в конфигурацията. Трябва да се определели дали подобна промяна не е довела до появата на нови бъгове. Една от основните причини, поради която се правят тези тестове, е да се определи дали промяната в една част на системата не засяга и други части от системата.

## Integration Test /Интеграционно тестване/

Целта на тестовете е да се проверят основните елементи на дизайна. Тестовете са конструирани, така че да се провери дали всички компоненти си взаимодействат правилно.

# Какво ще се тества

## Преглед на планираните тестове

### Преглед на функциите , които ще се тестват

* Регистрация на физическо лице, чрез уеб форма;
* Регистрация юридическо лице;
* Извършване на покупко-продажба;
* Влизане в системата;
* Управление на профил;
* Управление на количка;
* Комуникация на системата с потребителя–автоматизирани имейли.

### Преглед на допълнителни функции, които ще се тестват

* Регистрация на физическо, чрез Facebook;
* Справки.

# Подходи при тестване

## Измерване степента на тестване

Степен на успеваемост при тестване

* Покритие на кода;
* Резултати;
* Рискове при проектирането;
* Рискове при тестването;
* Проблеми при тестването.

## Идентифициране на тестовете

Документи нужни за определяне на тестовете са:

* Главен план на проекта;
* Списък на рисковете;
* Спецификация на изискванията и допълнителните изисквания
* Визия;
* Дизайн модел;
* Бизнес модел;
* Модел на потребителските случай.

# Критерий за вход и изход

## Главен план за тесване

### Входящ критерий на главния план за тестване

При наличие на готова финкционалност от Системата за електронна търговия – BBay.

### Изходящ критерий на главния план за тестване

След изпълнението нa всички функционалности и тестването им се сформира съответния артефакт.

### Критерии за спиране и възобновяване

Причина за спиране и повторното възобновяване на тестване може да е спирането на проекта. Ако разработката се възобнови тестването също ще продължи.

# Продукти от тестването

* Главен план за тестване;
* Тестови резултати от всяка фаза;
* Тест скриптове.

# Процес на тестване

След осъществяването на функционалност от системата във фазата „Детайлизиране“ започва тестването.Ще се тестват прототипи на системата, като всички функционалности ще се приоритизират и изпълняват в различни итерации в зависимост от сложността и приоритета им.

## Изисквания към средата

### Базов хардуер на системата

|  |  |
| --- | --- |
| **Системни ресурси** | |
| **Ресурс** | **Ресурс** |
| Сървър за база данни  Мрежа  Сървър  Бази данни |  |
|  |
| PostgreSQL |
| bbay |
| Драйвър | Postgresql-jdbc4 |
| Компютри за тестване: Lenovo N200 и десктоп машина | |
| Конфигурация лаптоп Lenovo N200 | OS: Windows Professional 7 64-bit  Processor: Intel Core 2 Duo T7300 (2GHz)  RAM: 1GB  Video: nVidia GeForce Go 7300  Display: 15.4in TFT,1280x800   |  |  | | --- | --- | |  |  | |
| Конфигурация десктоп машина | OS: Windows Professional 7 64-bit  Processor: Intel Core I5 3570K (3.4GHz)  RAM:8 Gb  Video: AMD ATI Radeon 7770  Dislplay: Asus 21inc |
| Включва специални конфигурационни изисквания | VPN връзка към vpn.uni-ruse.bg:   * User name: mse2012 * Password: @iv1213   Инсталиран специализиран софтуер:   * Eclipse JEE JUNO SR2 Win32 * JBoss 7.1.1. Final * Maven 2.0.6 * SVN - Subclipse |

## Базови софтуериелементи на тестовата среда

| **Име на софтуерния елемент** | **Версия** | **Тип и други** |
| --- | --- | --- |
| Mozilla Firefox | 9 | Web браузър |
| Opera | 10.50 | Web браузър |
| Chrome | 15 | Web браузър |
| Internet Explorer 8 | 8 | Web браузър |

## Продуктивност и проддръжка

За тесването ще се използват инстоментите описани във TFr таблицата.

## Отговорности при тестването

|  |  |
| --- | --- |
| Човешки ресурс | |
| Роля | Отговорности |
| Test Manager /Тест ръководител/ | Осигурява управлението.  Отговорности:   * Планиране и логистика; * Идентифициране на мотиваторите; * Определяне на подходящи ресурси; * Изготвяне на мениджмънт отчети.   Повишаване ефикасността на тестовете |
| Test Analyst /Тест аналитик/ | Набелязва и определя съответните тестове, които ще се изпълнят.  Отговорности:   * Идентифициране на тестовете; * Дефиниране на тестовите детайли; * Определяне на резултатите от тестването; * Правене промени по документите за тестването.   Измерване качеството на тестването |
| Tester /Тестер/ | Определя техническия подход за определяне на тестовете.  Отговорности:   * Определяне подхода при тестовете; * Определяне архитектурата на автоматизираните тестове; * Проверка на техниките за тестване; * Определяне на елементите, които ще се тестват.   Структуриране на тест имплементацията |
| Test System Administrator | Осигурява тестовата среда  Отговорности:   * Администриране на управленската тестова среда;   Инсталиране и поддържане на конфигурациите на тестовата среда и тестовите лаборатории. |
| Designer /Дизайнер на тестове/ | Определя операциите, атрибутите и асоциациите на тестовите класове  Отговорности:  Дефиниране на тестовите класове, които се изискват от тест екипа, за да се изпълнят тестовите изисквания |
| Implementer /Разработчик/ | Имплементира тестовите класове  Отговорности:  Създаване на тестовите компоненти дефинирани от тест дизайнера. |

## Обучение на тестери

## Рискове за главния план за тестване

| **Риск** | **Стратегия за ограничаване** | **Разходи** |
| --- | --- | --- |
| Недостатък на човешки ресурси | Наемане на достатъчно количество хора за тестването на настоящия проект | Забавяне при тестването. Разходи за наемане на още персонал |
| Недостатък на времето | Тестват се тези тестови случаи с най-висок приоритет. Подреждат се във възходящ ред тестовите случаи. | Наемане на още хора, за да може да се изтества колкото се може повече софтуер за определеното време |
| Голям брой бъгове и забавяне в отстраняването им | Превенция за откриване на грешките в ранен етап | Забавяне на тестването |
| Неправилно подбран продукт за тестване | Предварително обстойно запознаване | Забавяне на процеса по тестване и некачествено изпълнение |