**План по внедрению процесса тестирования в разработку**

|  |  |  |
| --- | --- | --- |
|  | | |
| **Версия** | **Составил** | **Изменения** |
| **1.0** | **Расулзода М** |  |

**Общая схема процесса тестирования (к чему будем стремиться)**

Релиз кандидат 2

Релиз кандидат N

Релиз кандидат 1

Автоматические тесты (unit tests, selenium, PyTest etc.)

Сервер непрерывной интеграции (CI)

Основная ветка в хранилище (trunk)

…

Выбор релиз кандидатов

Нет Есть

Отчет по релиз кандидату

Отчет по тестам

Отчет по тестам

Отчет по тестам

Нагрузочное тестирование

Системное тестирование

Регрессионное тестирование

Рис 1.

Релизная ветка в хранилище (tags)

Примечания:

Штрих по контуру – взаимодействия никакого нету, т.е. данные не заносятся и не редактируются.

Без штриха в контуре – взаимодействие может включать, какое либо вмешательство со стороны пользователя (тестировщик, разработчик, руководитель).

**Описание процесса**

– Основная ветка в хранилище (trunk)

В данной ветке хранятся текущие разработки. При каждом новом коммите, запускается сборка этого коммита. Каждая сборка после успешной компиляции проходит авто-тесты. И если все авто-тесты пройдены, то отмечается как релиз-кандидат.

– Сервер непрерывной интеграции (CI)

Какой именно сервер надо выбирать исходя из следующих критерий:

– Большинство разработчиков работали с данной системой

– Имеет поддержку необходимых инструментов

* Поддерживает Python и фреймворки к нему
* Является развиваемым проектом и поддерживаемым
* Внедрение не требует высоких затрат

Задачи сервера интеграции будут такими же как и большинство стандартных задач (сборка, запуск тестов, хранение результатов сборки и т.п.)

– Автоматические тесты, selenium и модульные тесты.

На текущий момент работу по настройке, запуску авто-тестов, а также написанию тестов на Selenium могу взять на себя. По написанию модульных тестов, пока силами разработчиков. До того момента пока научусь сам писать.

– Выбор релиз кандидатов

Соответствует или нет сборка требованиям релиза-кандидата. Тестировщик смотрит, насколько данная сборка соответствует требованиям и насколько критичны баги, которые имеются. Кто может стать релиз-кандидатом решается не только тестировщиком.

– Регрессионное тестирование

Проверка исправленных багов, а также, не повлияли ли эти исправления на другие функционалы

– Системное тестирование

Тестирование всех функциональных и не функциональных требований. Примеры: отображение сайта на разных браузерах, реагирование кнопок на разные исходные ситуации, проверка быстродействия API при обращении из других регионов или стран и т.д.

– Нагрузочное тестирование

Проверка системы на стрессы, большое количество запросов в секунду, скорость обработки данных и т.д.

**Этап 1**

Основная ветка в хранилище (trunk)

Сервер непрерывной интеграции (CI)

Ручное тестирование (включает все виды тестирования)

|  |  |
| --- | --- |
| **Задача** | **Примерное время** |
| Внедрение порядка и версионности (если они отсутствуют) | 0-2 недели |
| Выбор системы непрерывной интеграции | 1-2 недели, включает сбор и анализ возможностей того или иного CI |
| Внедрение системы непрерывной интеграции | 4-8 недель, зависит сильно от количества компонентов системы и отвлетленных веток (branches) |
| Настройка доступа и прав | 1-2 недели, здесь проблемы могут быть организационные, поэтому беру максимум |
| **Общее** | **Минимум** – 6 нед. **Максимум** – 14 недель |

**Этап 2**

Ручное тестирование (включает все виды тестирования)

Сервер непрерывной интеграции (CI)

Основная ветка в хранилище (trunk)

Автоматические тесты (unit tests, selenium, PyTest etc.)

|  |  |
| --- | --- |
| **Задача** | **Примерное время** |
| Написание тестов на Selenium | ~ Зависит от объема, который требуется автоматизировать. Из текущей версии сайта (2-3 недели) |
| Настройка и запуск существующих тестов | 0,5 – 1 неделя, тоже зависит от того, сколько их. |
| **Общее** | **Минимум** – 2 нед. **Максимум** –4 недели |

**Этап 3**

Отчет по тестам

Отчет по тестам

Отчет по тестам

Нагрузочное тестирование

Системное тестирование

Регрессионное тестирование

Автоматические тесты (unit tests, selenium, PyTest etc.)

Сервер непрерывной интеграции (CI)

Основная ветка в хранилище (trunk)

|  |  |
| --- | --- |
| **Задача** | **Примерное время** |
| Разделение видов тестирования и документации | Все время от 1-го этапа и до Continuous Delivery (4-й этап) |
| Интеграция инструментов для тестирования (TestRail, HP QC/ALM или другое) | ~ Сложно оценить время. Сильно зависит от объема нужной документации и того, что можно документировать |

**Этап 4**

(См. Рис 1)

|  |  |
| --- | --- |
| **Задача** | **Примерное время** |
| Внедрение Continuous Delivery | ~ |
| Определение того, что можем считать релиз-кандидатом | ~ |

Итог: Полный объем времени на текущий момент определить трудно. Можно предположить время только на этапы 1-2.

**Минимум** – 8 нед. **Максимум** – 18 недель