**План индивидуального развития младшего специалиста по качеству ПО Дудник Виктор до 01.12.2015**

**Что нужно уметь до 01.12.15:**

1. Подготовка тестовых данных перед началом тестирования на основе чек - листа или тест - кейсов. Использование генерации данных перед началом тестирования. Использование iMacro, Selenium, jmeter для автоматизации создания данных.
2. Проверка готовности окружений развернутых для тестирования. Умение провести Sanity check.
3. Скриптовое тестирование (функциональное или регрессионное). Проход тестов по тестовой документации (тест-кейсы), проход тестов по чек листу, запуск авто-тестов, запуск нагрузочных тестов, анализ результатов, написание отчета, обработка статистики запуска, оформление результатов, идентификация неполного покрытия тестами тестируемого функционала, информирование руководителя или наставника о необходимости обновления скриптов ручного, автоматического или нагрузочного тестирования. Самостоятельное обновление скриптов нагрузочного или ручного тестирования. Выполнение обновленных тестов.
4. Выполнение тестов, по проверке исправленных несоответствий (дефекты, баги), сравнение ER и AR, коррекция шагов воспроизведения, отчет руководителю. Проверка возможной регрессии. Информирование руководителя или наставника о необходимости обновления скриптов ручного, автоматического или нагрузочного тестирования. Самостоятельное обновление скриптов нагрузочного или ручного тестирования. Выполнение обновленных тестов.
5. Проведение верификации дефекта при отсутствующих или не очевидных шагах воспроизведения. Проведение исследовательского тестирования. План исследовательского тестирования должен быть согласован с руководителем или со старшим специалистом. Информирование руководителя о необходимости обновления тестовой документации или скриптов автоматизированного тестирования. Обновление тестовой документации. Обновление скриптов нагрузочного тестирования.
6. Исследование внештатных ситуаций при выполнении шагов скрипта ручного тестирования, или причин отклонения от ожидаемого результата. Исследование причин падения авто-тестов, скриптов нагрузочного тестирования, завершившихся с большим количеством ошибок. Обновление скриптов ручного или нагрузочного тестирования.
7. Тест-дизайн уровня проекта. Определение цели тестирования. Определение объекта тестирования. Определение видов тестирования (приемочное, установочное, альфа и бета тестирование). Определение входных данных. Выбор видов тестирования и их применения по отношению к объекту тестирования. Составление плана тестирования.
8. Тест-дизайн уровня проекта. Разработка чек - листов для тестирования небольших приложений.
9. Тест-дизайн уровня проекта. Идентификация всех значений, которые вводятся участниками в сценарии использования системы. Выделение классов эквивалентности значений каждого типа входных данных. Построение тестовых случаев, в которых сочетаются одна перестановка значений с необходимыми внешними ограничениями. Написание программ для автоматизированного тестирования (при необходимости). Изучение плана тестирования и выявление приоритетных областей покрытия тестовыми случаями. Построение таблиц, в которые помещен список комбинаций значений из различных классов эквивалентности.
10. Поддержка тестовой документации по проектам в актуальном состоянии после каждого функционального тестирования или проверки исправленного несоответствия.
11. Создание авто-тестов на рабочем framework, по созданным сценариям.
12. Анализ тестируемой системы с выделением критичных к нагрузке узлов, идентификацией сценариев использования представляющих высокую нагрузку для системы.
13. Написание профильных и сценарных скриптов нагрузочного тестирования для приложений с различной web-архитектурой.
14. Идентификация и создание данных необходимых для работы скриптов нагрузочного тестирования.

**Теоретические знания.**

|  |  |  |
| --- | --- | --- |
| **№** | **Знание** | **Срок** |
| 1. | Архитектура web-приложений. Знать основные технологии. Специфика тестирования web-приложений, развернутых на серверах, построенных по различным технологиям. | 30.09.2015 |
| 2. | Знание основных 7 техник тест-дизайн: разбиение на классы эквивалентности, использование граничных значений, использование таблицы решений, использование таблицы состояний, тестирование по области значений, использование комбинаторных методов в тест-дизайн. | 30.09.2015 |
| 3. | Знание основ нагрузочного тестирования. Базовые понятия – нагрузочная точка, модель нагрузки, профиль нагрузки. Метрики, снимаемые по результатам прогона авто-тестов с серверов приложений, серверов баз данных: их характеристика и влияние на производительность. | 30.09.2015 |
| 4. | HTTP протокол. Знание стандарта. Структура протокола. Заголовки запросов, ответов. Передача контента. Кэширование. | 31.10.2015 |
| 5. | Специфика проведения нагрузочного тестирования для web-приложений с различной архитектурой. | 30.11.2015 |
| 6. | Знание стандарта качества ПО ISO-9126: основные критерии и факторы качества. | 30.11.2015 |

**Практические навыки**

|  |  |  |
| --- | --- | --- |
| № | Навык | Срок |
| 1. | Уверенное использование основных техник тест-дизайн: разбитие на классы эквивалентности, контроль граничных значений, использование таблицы решений. | 30.09.2015 |
| 2. | Умение правильно построить тест-сьют с выделением тест-кейсов для позитивных и негативных сценариев, разных типов тестирования. | 30.09.2015 |
| 3. | Умение разработать набор сценариев, имитирующих поведение пользователей, для создания сценарных скриптов нагрузочного тестирования. | 31.10.2015 |
| 4. | Уметь развернуть и настроить необходимый инструментарий для разработки и запуска авто-тестов для тестирования web-приложений. | 31.10.2015 |
| 5. | Создание скриптов нагрузочного тестирования: профильных и сценарных. Умение создать необходимые данные, используемые скриптами нагрузочного тестирования. | 30.11.2015 |
| 6. | Умение работать с инструментами мониторинга использования ресурсов сервером (облачные решения - DataDog, NewRelic или локальные решения – Sysstat, Task Manager). | 30.11.2015 |
| 6. | Умение использовать usability check list | 30.11.2015 |

**Рекомендуемая литература:**

Документы и литература по тестированию, которую стоит почитать:

1. Канер, Фолк. Тестирование программного обеспечения.
2. Lee Copeland. Practitioner’s guide to software test design.
3. Rfc2616. HTTP protocol standard.
4. Performance testing guide (Microsoft) 2007.
5. Винниченко И.В. Автоматизация процессов тестирования