

Домашняя работа (Занятие 4)

Имя	Андрей Н.	Предмет	QA-3 Testing
Преподаватель	Виталий З.	Период обучения	04-10/2023
Дата	10.05.2023		
Тема	Анализ существующих моделей и методов разработки. Роль и место QA в процессе разработки программного обеспечения, тестирование и QA		

Введите инструкции для теста. Например, укажите, что учащиеся должны написать короткое описание к ответу на каждый приведенный вопрос.

1. Выберите и изучите одну из моделей разработки ПО, которую мы не рассматривали на уроке. Напишите эссе на пять-семь предложений. Укажите, почему Вы выбрали именно эту модель, какие у неё достоинства/недостатки. Желательно написать о модели разработки ПО, с которой Вы бы хотели работать в реальности и обосновать почему.

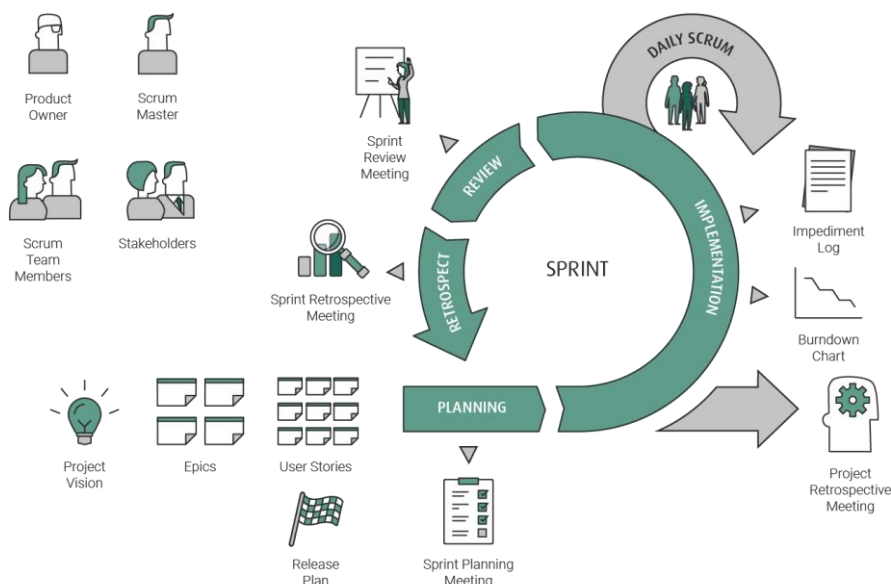


Рисунок. SCRUMBAN методология

В качестве модели для разработки ПО, предлагается рассмотреть SCRUMBAN методологию, которая включает в себя две распространенные стратегии Agile: SCRUM и KANBAN. Первоначально SCRUMBAN был разработан, чтобы помочь командам перейти от Scrum к Kanban или наоборот. Если у команды есть опыт работы с одной

стратегией по сравнению с другой, этот метод может помочь им постепенно перейти к другой методологии.

Возможности которые имеет SCRUMBAN по заимствованных из Scrum:

- Итерации процесса происходят через определенные промежутки времени, обычно в конце спринтов, когда команда проводит ретроспективы спринтов и собрания по обзору спринтов.
- Работа расставляется по приоритетам в зависимости от сложности задачи и спроса на продукт.
- Вся команда соглашается и соглашается с тем, что означает «выполнено», поэтому все знают, что значит выполнить задачу. Это означает, что конечные результаты четко очерчены и определены.

Возможности которые имеет SCRUMBAN по заимствованных из Kanban:

- Использует четкий список задач, которые необходимо выполнить. Как только член команды начинает работать над задачей, он «перетаскивает» эти задачи из журнала невыполненных работ в свою текущую рабочую нагрузку.
- Существуют жесткие ограничения на количество задач, которые в настоящее время выполняются, чтобы предотвратить переутомление команды.
- Задачи визуально представлены в виде карточек, которые проходят различные этапы процесса на доске Канбан.

Для SCRUMBAN характерны и специфические возможности:

- В Scrumban нет командной иерархии. Это означает, что каждый человек в команде разработчиков имеет такую же возможность, как и другие, принимать решения и делать выбор. Это также означает, что в группе нет явного лидера — скорее, команда полностью самоуправяема.
- Проекты Scrumban не обязательно должны иметь крайний срок. Спринты часто выполняются с двухнедельными интервалами, поэтому члены команды могут просто сосредоточиться на этих конкретных задачах спринта, пока не придет время просмотреть и повторить. Это делает Scrumban хорошим выбором для очень долгосрочных проектов или проектов с неоднозначной целью. Также его стоит применить для постоянного обслуживания ПО, которое не имеет четких временных рамок.

Преимущества	Недостатки
Может помочь сэкономить время : использование метода Scrumban — хороший способ начать отслеживать выполняемую работу. Используя метод Scrumban, вы предотвращаете дублирующую работу или трату время на задачи, которые не соответствуют цели конкретного спринта.	Отсутствие руководства может вызвать затруднение: в то время как независимость и автономия могут мотивировать одну команду, отсутствие надзора может вызвать замешательство и дезорганизацию в другой. Важно знать, что то, что работает для одной команды, может не работать для другой. Чтобы найти правильную методологию управления проектами, нужно выяснить, что работает для конкретной команды.

Отлично подходит для долгосрочных или крупномасштабных проектов : поскольку Scrumban позволяет вносить небольшие изменения в течение больших промежутков времени. Это делает его отличным фреймворком для долгосрочных проектов, поскольку потребности проекта будут меняться с течением времени. По мере изменения потребностей Scrumban помогает повторять и улучшать процессы, чтобы не отставать от этих изменений.	Scrumban — относительно новая методология : поскольку Scrumban — более новая методология, устоявшихся процессов не так много. Процесс одной команды Scrumban может сильно отличаться от процесса другой команды, и отчасти это связано с отсутствием стандартизированной структуры, как в Scrum или бережливом управлении проектами.
Члены команды обладают большей независимостью : если команда хочет большей автономии, методология Scrumban может стать хорошим началом. Метод Scrumban дает членам команды возможность принимать решения и расставлять приоритеты в работе по своему усмотрению, а не просто выполнять работу, назначенную скрам-мастером или владельцем продукта.	Менеджеры проектов имеют меньше контроля: если у вас есть менеджер по продукту или менеджер проекта, который очень практичен, эта методология может не сработать. В Scrumban нет конкретных ролей в команде разработчиков. Это означает, что у всех есть одно и то же агентство, чтобы выбрать то, что они считают правильным решением для спринта.

Данная модель разработки ПО подходит, для меня больше всего т.к. я хочу работать в долгосрочных или крупномасштабных проектах с большим количеством профессионалов.

Источник: <https://asana.com/resources/scrumban>

2. Опишите обязанности инженера QA. (Внимание: не тестировщика, а инженера QA.)

Должностные обязанности инженера по обеспечению качества.

Цель обеспечения качества состоит в том, чтобы определить и поддерживать набор требований и стандартов в разработке программного обеспечения. Это помогает гарантировать, что каждая система и этап разработки соответствуют отраслевым стандартам и требованиям конечных пользователей (и бизнеса). Часто процессы обеспечения качества следуют международным стандартам ISO 9000:2015. И эффективная и действенная система контроля качества дает разработке много преимуществ. Строгий контроль качества программного обеспечения повышает имидж и репутацию вашей компании. Кроме того, это укрепляет доверие клиентов к использованию программного обеспечения. Самое главное, это дает продукту преимущество перед конкурентами.

Таким образом обеспечение качества (QA) отличается от контроля качества (QC).

Контроль качества (QC) включает в себя действия по тестированию, которые сосредоточены на самом продукте. Эти действия осуществляются для проверки и проверки соответствия программного обеспечения установленным требованиям.

С другой стороны, обеспечение качества (QA) ориентировано как на продукт, так и на процесс. Помимо изучения качества программного обеспечения, QA также изучает стандарты, которым следуют на каждом этапе его разработки.

Ниже приведем пример основных обязанностей QA инженера:

- ❖ Установите цели процесса обеспечения качества, чтобы гарантировать, что результат приведет к выполнению всех требований к продукту.
- ❖ Планирование мероприятий по тестированию в соответствии с целями и создание стратегии на основе графика и других особенностей проекта. Проведение мероприятий по тестированию, будь то ручное или автоматизированное тестирование качества, на соответствующих этапах разработки для устранения ошибок и недостатков.
- ❖ Работа с разработчиками, тестировщиками и другими заинтересованными сторонами для устранения сбоев и ошибок тестирования.
- ❖ Коммуникация с руководителями проектов и другими заинтересованными сторонами о системах обеспечения качества, мероприятиях и прогрессе.
- ❖ Обеспечение правильности и полноту документации по управлению качеством. Контролирует выполнение процессов и планов обеспечения качества (планы инспекционных испытаний и планы контроля качества)
- ❖ Организовать периодические собрания команды для обеспечения согласованности с процессом обеспечения качества и решения проблем с помощью инструментов обеспечения качества.
- ❖ Сбор комментариев и отзывы для оптимизации систем и процессов
- ❖ Поддержка автоматизированных процедур тестирования
- ❖ Отслеживать производительность и обнаруживать уязвимости системы с помощью инструментов и систем отслеживания ошибок.

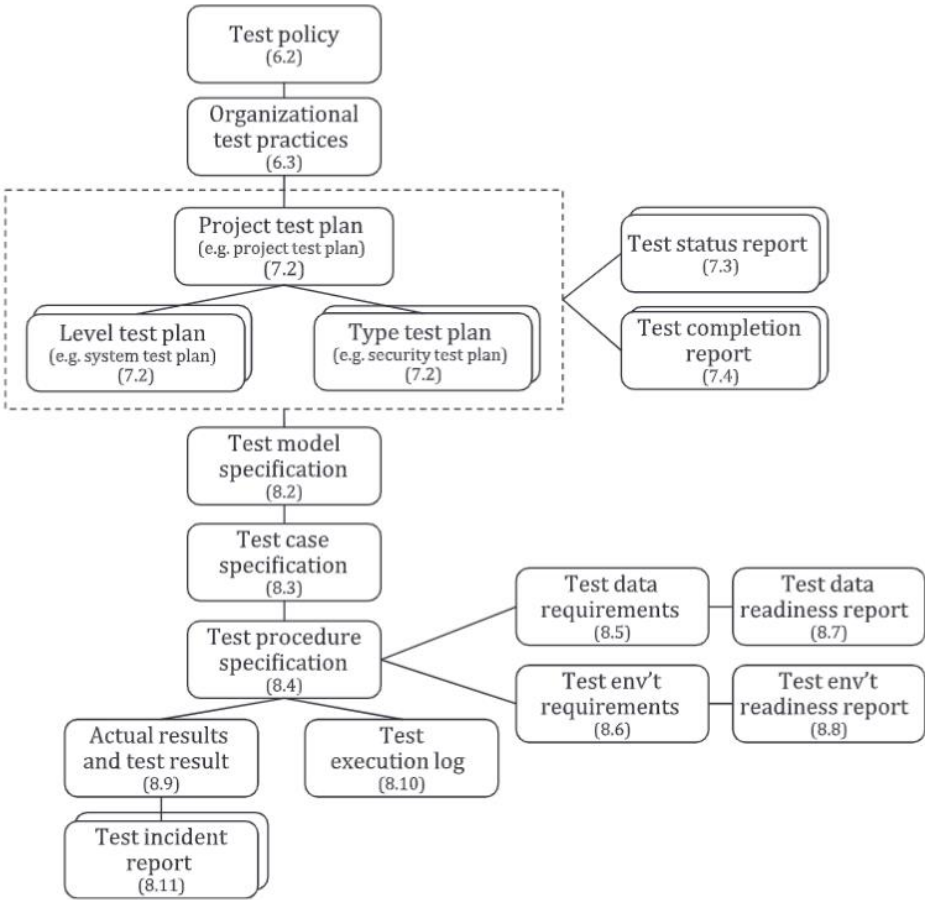


Figure 1 — Overview of test documentation