

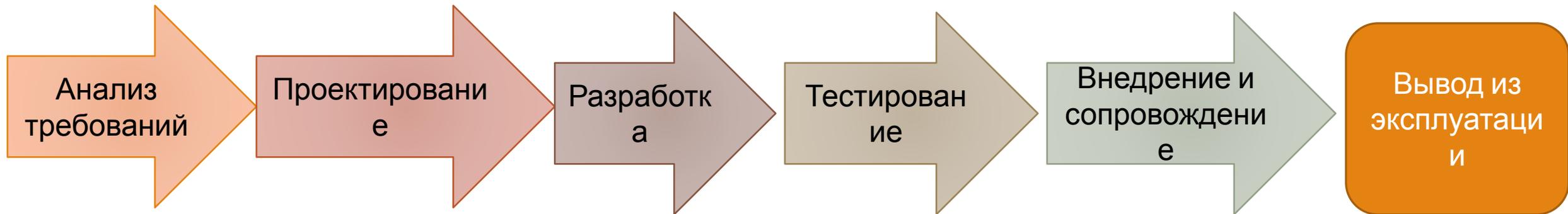
Урок 2. Модели разработки ПО. Agile, Scrum

NORDIC IT SCHOOL

О чем мы сегодня поговорим

- Модели разработки ПО
 - Работа тестировщика в условиях самых распространенных моделей разработки (Waterfall, Scrum)
- Тестирование.
 - Что такое тестирование?
 - Функции тестирования?
 - Для чего нужен и чем занимается тестировщик на проекте?
 - Принципы тестирования
- Практические задания
 - Тестирования формы ввода имени
 - Тестирование элементов дизайна

Жизненный цикл ПО

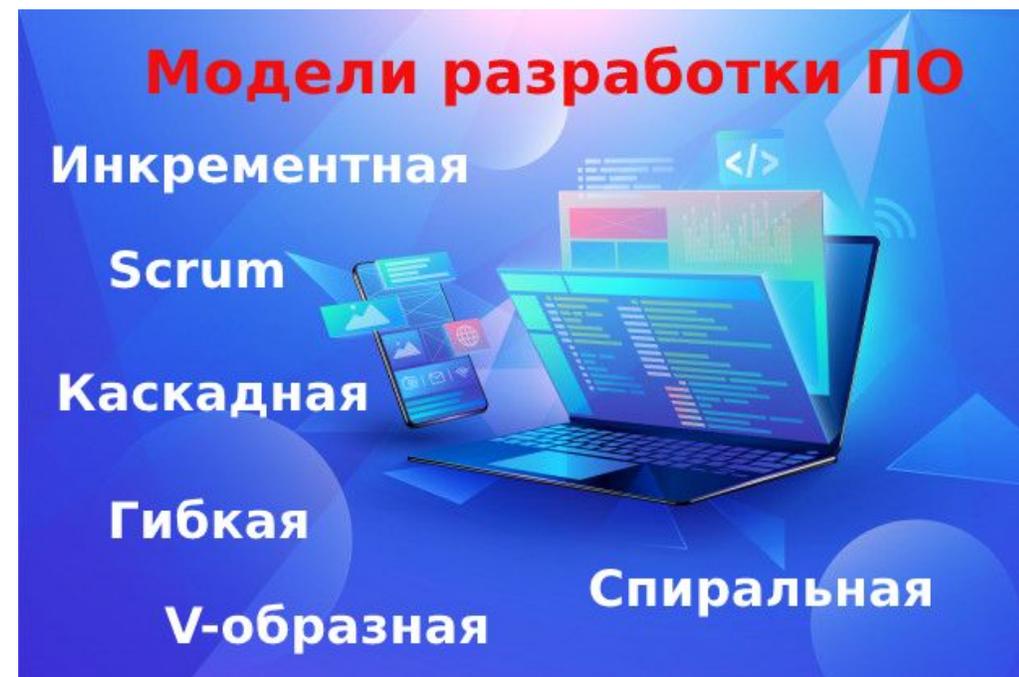


Роли в ИТ

- Менеджер проекта
- Менеджер продукта
- Аналитик
- Дизайнер
- Разработчик
- Специалист по тестированию
 - Тестировщик
 - QC (Quality Control)
 - QA (Quality Assurance)
- Scrum master (опционально)

Модели разработки ПО

Модель жизненного цикла ПО – структура, определяющая последовательность выполнения и взаимосвязи процессов, действий и задач на протяжении жизненного цикла. Модель жизненного цикла зависит от специфики, масштаба и сложности проекта и специфики условий, в которых система создается и функционирует.



Каскадная («Водопад») модель

Каскадная модель (Waterfall model) – модель процесса разработки программного обеспечения, жизненный цикл которой выглядит как поток, последовательно проходящий стадии анализа требований, проектирования, реализации, тестирования, интеграции и поддержки.



Инкрементальная модель

Инкрементальная модель (Iterative model) – подразумевает разработку программного обеспечения с линейной последовательностью стадий, но в несколько инкрементов (версии), т.е. с запланированным улучшением продукта за все время пока Жизненный цикл разработки ПО не подойдет к окончанию.



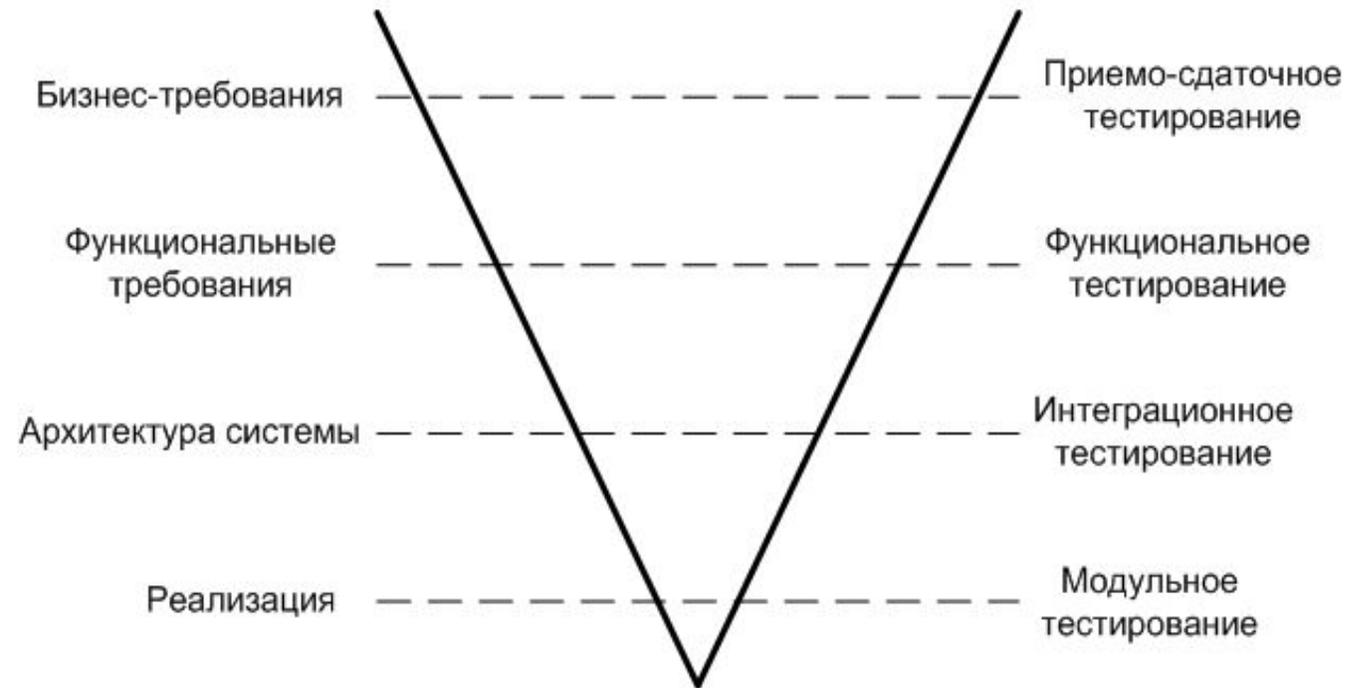
Спиральная модель

Спиральная модель (Spiral model) – на каждом витке спирали выполняется создание очередной версии продукта, уточняются требования проекта, определяются его качество и планируются работы следующего витка.



V - модель

V – модель – является расширением модели водопада и основана на ассоциации стадии тестирования для каждой соответствующей стадии разработки. Это очень строгая модель и следующий этап начинается только после завершения предыдущей фазы.



Гибкие методологии разработки ПО

Гибкая методология разработки (Agile software development, agile-методы) – серия подходов к разработке программного обеспечения, ориентированных на использование итеративной разработки, динамическое формирование требований и обеспечение их реализации в результате постоянного взаимодействия внутри самоорганизующихся рабочих групп, состоящих из специалистов различного профиля.

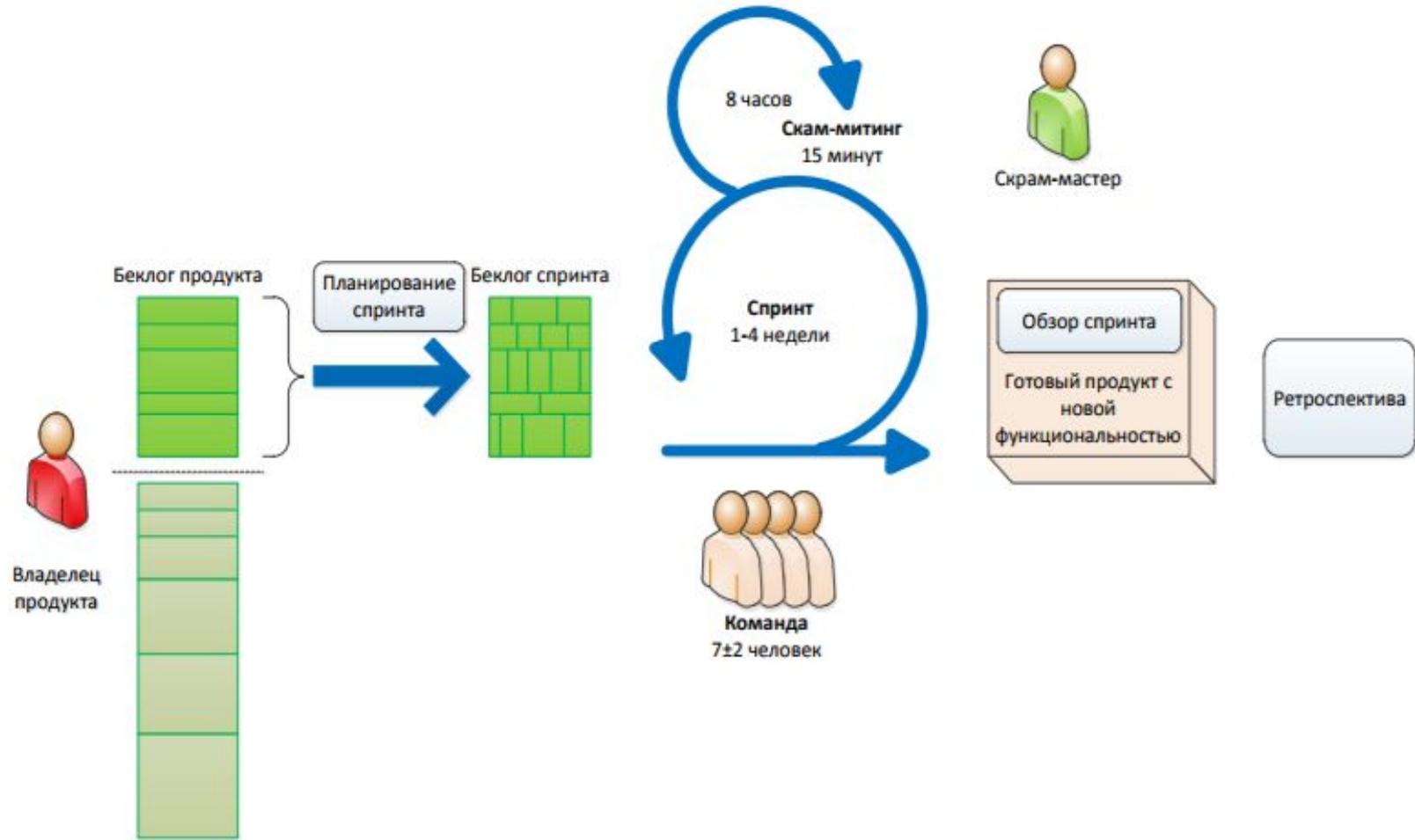


Ценности Agile

1. Люди и взаимодействие важнее процессов и инструментов.
2. Работающий продукт важнее исчерпывающей документации.
3. Сотрудничество с заказчиком важнее согласование условий контракта.
4. Готовность к изменениям важнее следования первоначальному плану.

Scrum

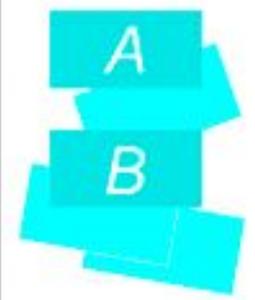
Scrum – это набор принципов, на которых строится процесс разработки, позволяющий в жёстко фиксированные и небольшие по времени итерации, называемые спринтами (sprints), предоставлять конечному пользователю работающее ПО с новыми возможностями, для которых определен наибольший приоритет. Возможности ПО к реализации в очередном спринте определяются в начале спринта на этапе планирования и не могут изменяться на всём его протяжении. При этом строго фиксированная небольшая длительность спринта придаёт процессу разработки предсказуемость и гибкость.



Kanban

Kanban – метод управления разработкой, реализующий принцип «точно в срок» и способствующий равномерному распределению нагрузки между работниками. При данном подходе весь процесс разработки прозрачен для всех членов команды. Задачи по мере поступления заносятся в отдельный список, откуда каждый разработчик может извлечь требуемую задачу

В планах	Разработка	Тестирование	Релиз	Готово!
5	3	2	3	
H	F	D	C	A
I	G	E		B
J				
K				



Переходим к тестированию

- Что такое тестирование?
- Цели тестирования?
- Роль тестирования в процессе обеспечения качества

Тестирование - это проверка, которая позволяет определить: соответствует ли реальное поведение программы ожидаемому, выполнив специально подобранный набор тестов.

Цель тестирования - получение информации о текущем состоянии продукта и готовности его к выпуску.

Тестирование- это не поиск ошибок!

Тестирование- это одна из составляющих процесса обеспечения качества продукта !

Для чего проводится тестирование ПО?

Цели тестирования ПО

- Оценка рабочих продуктов, таких как требования, пользовательские истории, проектирование и код
- Проверка, все ли указанные требования выполнены
- Проверка, завершен ли объект тестирования и работает, как ожидают пользователи и заинтересованные лица
- Создание уверенности в уровне качества объекта тестирования
- Предотвращение дефектов
- Обнаружение отказов и дефектов
- Предоставление заинтересованным лицам достаточной информации, позволяющей им принять обоснованные решения, особенно в отношении уровня качества объекта тестирования
- Снижение уровня риска ненадлежащего качества программного обеспечения (например, пропущенные сбои в работе)

Тестирование, контроль качества и обеспечение качества

QC (Quality Control) — Контроль качества продукта — анализ результатов тестирования и качества новых версий выпускаемого продукта.

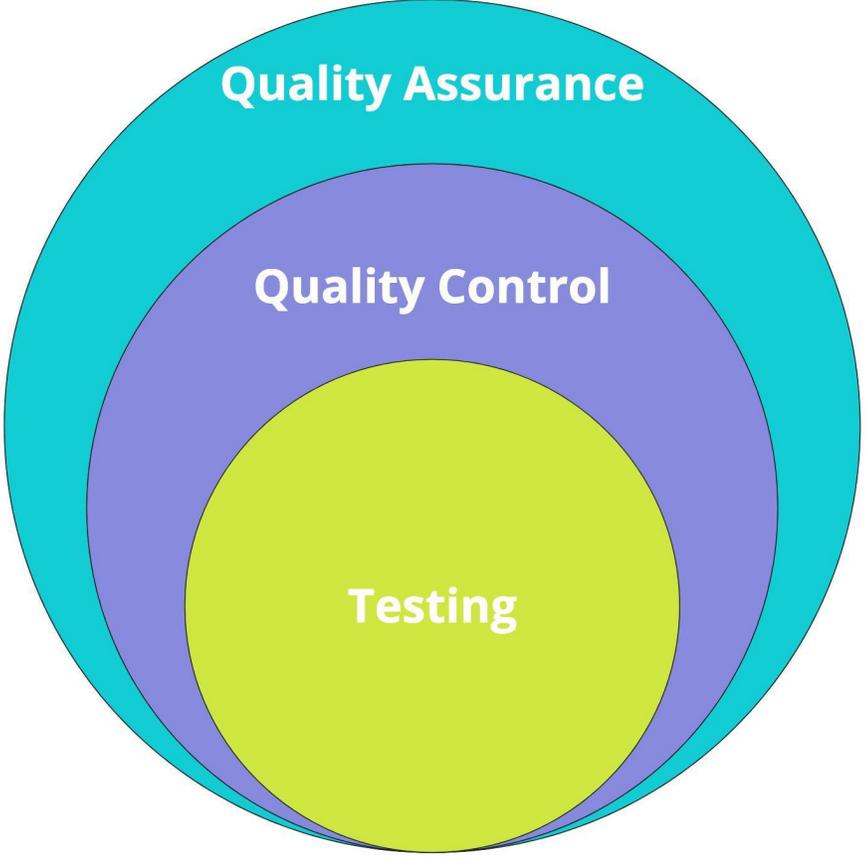
К задачам контроля качества относятся:

- проверка готовности ПО к релизу;
- проверка соответствия требований и качества данного проекта.

QA (Quality Assurance) — Обеспечение качества продукта — изучение возможностей по изменению и улучшению процесса разработки, улучшению коммуникаций в команде, где тестирование является только одним из аспектов обеспечения качества.

К задачам обеспечения качества относятся:

- проверка технических характеристик и требований к ПО;
- оценка рисков;
- планирование задач для улучшения качества продукции;
- подготовка документации, тестового окружения и данных;
- тестирование;
- анализ результатов тестирования, а также составление отчетов и других документов.



Принципы тестирования ПО

1. Обязательная часть тестирования – определение ожидаемого результата;
2. Программистам следует избегать тестирования их собственных программ (и участков кода);
3. Организациям, создающие программы, следует избегать тестирования их собственных программ;
4. Процесс тестирования должен включать в себя тщательную проверку результатов каждого теста;

Принципы тестирования ПО

5. Тест-кейсы должны быть составлены как для корректных и ожидаемых входных условий, так и для некорректных и неожиданных;
6. Исследование Системы на предмет того, что она не делает того, что должна, — лишь полдела. Вторая часть – разобраться в том, чего недолжного она делает;
7. Избегайте одноразовых тест-кейсов, только если сама программа не является одноразовой. Одноразовые тест-кейсы для одноразовых программ. В остальных случаях следует избегать таковых;

Принципы тестирования ПО

8. Не занимайтесь процессом тестирования с предубеждением, что вы не найдете ошибок;
9. Вероятность наличия ошибок в определенной части Системы пропорционально количеству уже найденных здесь ошибок;
10. Тестирование – это вызов вашим творческим и интеллектуальным способностям.
Тестирование – это невероятно творческое и интеллектуальное занятие.

Тренировочные задачи

- <http://prestashop.gatestlab.com.ua/ru/>

- Вы работаете в интернет – магазине женской одежды. Поступила задача – проверить правильность работы

1. добавления в корзину майки.
2. Просмотра корзины
3. Удаления майки из корзины
4. Свободное задание, связанное со сценариями выше

Давайте распишем по шагам действия тестировщика при работе над этой задачей.

- Задание:

1. Написание тест-кейса по задаче, которая попала к нам на тестирование
2. Проверка задачи по нашему тест-кейсу
3. Написание отчета об ошибке (если требуется)
4. Передача ошибки разработчику

Что мы сегодня узнали?

- С чем Вы выходите с занятия?
- Остались ли вопросы по занятию?
- Пожалуйста оцените наше сегодняшнее занятие

Спасибо за внимание!