



ТЕСТИРОВЩИК
ПРОГРАММНОГО
ОБЕСПЕЧЕНИЯ

КУРС «РУЧНОЕ ТЕСТИРОВАНИЕ»

2. SDLC, STLC

Жизненный цикл ПО, SDLC

Этапы SDLC

Жизненный цикл тестирования, STLC.
Преимущества и основные шаги.

ЖИЗНЕННЫЙ ЦИКЛ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Жизненный цикл программного обеспечения (ПО) — период времени, который начинается с момента принятия решения о необходимости создания программного продукта и заканчивается в момент его полного изъятия из эксплуатации.

Стандарты жизненного цикла ПО

ГОСТ 34.601-90

ISO/IEC 15288:2015 Systems and software engineering — System life cycle processes

ISO/IEC/IEEE 12207:2017 System and software engineering — Software life cycle processes

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.

SDLC

Software Development Life Cycle (SDLC) – это жизненный цикл программного обеспечения из набора типовых этапов, переход между которыми определяется моделью процессов разработки ПО.

Подробно цель, выходы, действия и задачи каждого из этих процессов описан в международном стандарте ISO/IEC 12207:2008 «System and software engineering — Software life cycle processes». Российским аналогом этого документа является ИСО/МЭК 12207-2010 «Информационная технология. Системная и программная инженерия. Процессы жизненного цикла программных средств», принятый Федеральным агентством по техническому регулированию и метрологии РФ в 2012 году взамен ГОСТ Р ИСО/МЭК 12207-99. Эти стандарты подробно описывают более 40 процессов ЖЦ ПО, от организационного обеспечения проекта до повторного применения программных средств, сгруппированные в 7 категорий. Эти регламенты очень детальные и универсальные, т.е. подходят для любого прикладного кейса. Обратной стороной этих достоинств является большой объем документов. Поэтому начинающим тестировщикам легче понять основные процессы ЖЦ ПО на примере этапов SDLC.

ЭТАПЫ ЖЦ ПО, SDLC



1. ПЛАНИРОВАНИЕ

На данном этапе владелец продукта отвечает на вопрос «Что нужно сделать?», а руководитель проекта – на вопрос «Как это сделать». Здесь также может принимать участие и бизнес-аналитик, чтобы понять потребности и перевести их в бизнес-требования.

2. АНАЛИЗ

Определение и документирование требований в виде ТЗ на разработку ПО и/или спецификации, что является основной рабочей обязанностью бизнес-аналитика.

- Создание спецификации требований к ПО (Software Requirements Specification (SRS))
- Определение требований к качеству (Software Quality Attributes (SQA))
- Проведение анализа рисков (Risk Analysis (RA))
- Создание планов валидации и верификации, тест план (V&V Plans, Test Plans)
- Определение критериев приемки ПО

3. ПРОЕКТИРОВАНИЕ

Определение дизайна и архитектуры ПО, а также другие особенности реализации, например, UI/UX-дизайн. Помимо ИТ-архитектора и дизайнеров, здесь работает системный аналитик, уточняя и дополняя требования от бизнес-аналитика. На практике обязанности системного и бизнес-аналитика может выполнять один человек.

4. РАЗРАБОТКА

Определение дизайна и архитектуры ПО (блок-схемы, диаграммы, макеты и прототип системы), а также другие особенности реализации, например, UI/UX-дизайн. Помимо ИТ-архитектора и дизайнеров, здесь работает системный аналитик, уточняя и дополняя требования от бизнес-аналитика. На практике обязанности системного и бизнес-аналитика может выполнять один человек.

5. ТЕСТИРОВАНИЕ И РАЗВЕРТЫВАНИЕ

Всестороннее и полноценное тестирование (оценка качества) системы на основании подготовленной тестовой документации в соответствии с требованиями. Повторяется до тех пор, пока не будут достигнуты критерии окончания тестирования.

После того, как тестировщики проверят представленный разработчиками результат на соответствие требованиям и отсутствие ошибок, к делу приступают DevOps-инженеры и администраторы, которые разворачивают продукт в реальной среде эксплуатации (production).

5. ТЕСТИРОВАНИЕ И РАЗВЕРТЫВАНИЕ

Развертывание:

- ПО устанавливается в рабочей среде – развертывается
- Проверяется наличие проблем по развертыванию, которые устраняются в ходе обнаружения
- Релиз системы
- Выполняется настройка рабочих окружений, конфигураций
- Проводится приемочное тестирование и передача продукта конечным пользователям

6. ПОДДЕРЖКА И СОПРОВОЖДЕНИЕ

Работающий продукт сопровождают администраторы и специалисты технической поддержки. Также здесь и на предыдущем этапе технические писатели создают руководства пользователя и администратора, а также прочую программную документацию, предусмотренную контрактом на этапе планирования.

- Обеспечение связи с пользователем
- Помощь в сборе и систематизации различных метрик – показателей работы ПО в реальных условиях
- Помощь в определении новых требований, в выявлении новых дефектов
- Обеспечение работы системы в соответствии со спецификациями и требованиями

ЖИЗНЕННЫЙ ЦИКЛ ТЕСТИРОВАНИЯ. ОСНОВНЫЕ ШАГИ

Тестирование ПО представляет из себя процесс проверки того, насколько поведение реального ПО соответствует требованиям и ожиданиям. Проводимые для этого тесты основаны на задокументированных требованиях заказчика.



ЖИЗНЕННЫЙ ЦИКЛ ТЕСТИРОВАНИЯ (STLC)

Тестирование ПО представляет из себя процесс проверки того, насколько поведение реального ПО соответствует требованиям и ожиданиям. Проводимые для этого тесты основаны на задокументированных требованиях заказчика.



РОЛЬ ТЕСТИРОВАНИЯ В ЖИЗНЕННОМ ЦИКЛЕ РАЗРАБОТКИ ПО

Процесс тестирования ПО тесно связан непосредственно с процессом разработки. Жизненный цикл разработки состоит из следующих этапов:

1. Анализ требований
2. Дизайн
3. Разработка
4. Тестирование и дебаггинг
5. Эксплуатация и поддержка

Как показано в списке выше, мы должны провести тестирование на четвертом шаге жизненного цикла. Но обычно в случае, если нашей главной целью является получить высококачественное ПО и минимизировать затраты на исправление багов, мы можем проводить тестирование уже на стадии анализа требований. Чем раньше вы приступите к тестам, тем лучших результатов добьетесь.

ПРЕИМУЩЕСТВА ПРОВЕДЕНИЯ ТЕСТОВ НА КАЖДОМ ЭТАПЕ ЖИЗНЕННОГО ЦИКЛА ПО

Первый этап. Анализ требований

- Требования к конечному продукту обычно формулируются заказчиком или менеджером проекта. Эти требования могут быть как функциональными, так и нефункциональными. Они формируются в процессе общения с заказчиком или анализа стандартов и нормативной документации. Зачем же необходимо проводить тесты ПО на этом этапе жизненного цикла и какие преимущества это может нам принести?
- - QA может обнаружить ошибки не только в самом продукте, но также и в документации. Как вы можете представить, в этом случае исправление ошибок обойдется гораздо дороже в сравнении с подходом, который предусматривает включение тестов в самые ранние этапы жизненного цикла ПО, такие как фаза анализа требований. Тщательным образом проанализировав требования, вы можете собрать информацию, которая поможет вам оптимизировать процесс работы над проектом с самых первых дней.
- Главная задача такого подхода — убедиться в том, что требования заказчика были правильно интерпретированы и остаются корректными, понятными и последовательными. Важно отметить, что ясная и точная документация помогает выбрать правильные цели для процесса тестирования.

ПРЕИМУЩЕСТВА ПРОВЕДЕНИЯ ТЕСТОВ НА КАЖДОМ ЭТАПЕ ЖИЗНЕННОГО ЦИКЛА ПО

Второй этап. Процесс дизайна

Этот этап подразумевает проверку уже созданных прототипов на предмет их корректности и соответствия ожиданиям заказчика. Более того, проверка удобства в использовании также должна быть проведена на этом этапе. Также следует начать создание тестовой документации для данного проекта. Эта задача включает в себя подготовку плана тестирования, тест-кейсов, use-кейсов, а также другой документации по требованию заказчика.

Процесс тестирования ПО на этом этапе обеспечивает способность проникновения в суть продукта и понимание ее соответствия требованиям. Важным является точное понимание задач, стоящих перед отделом QA на протяжении всего жизненного цикла разработки.

ПРЕИМУЩЕСТВА ПРОВЕДЕНИЯ ТЕСТОВ НА КАЖДОМ ЭТАПЕ ЖИЗНЕННОГО ЦИКЛА ПО

Третий этап. Разработка

В течении этапа разработки важно провести модульное, интеграционное и системное тестирование.

В самом начале этого шага разработки проводится модульное тестирование. Этот процесс представляет собой проверку отдельного модуля системы или функционала.

Интеграционное тестирование проводится после того, как несколько модулей объединены вместе как отдельная часть приложения.

В дальнейшем в процессе разработки все больше и больше модулей объединяются воедино.

После того, как разработка закончена, наступает время подготовки к системному тестированию. Эта стадия жизненного цикла разработки ПО подразумевает общий тест системы на предмет интеграции ее компонентов. Это значит, что в случае, если система состоит из различных модулей, мы должны проверить, насколько хорошо или насколько плохо каждый из них работает внутри системы. Более того, на этом этапе важно произвести тестирование пользовательского интерфейса.

ПРЕИМУЩЕСТВА ПРОВЕДЕНИЯ ТЕСТОВ НА КАЖДОМ ЭТАПЕ ЖИЗНЕННОГО ЦИКЛА ПО

Четвертый этап. Процесс тестирования и дебаггинга

- На этом шагу должны быть проведены тесты независимо от того, проводились ли они на предыдущих этапах. Должны быть проведены полное функциональное тестирование и тестирование пользовательских интерфейсов, а все обнаруженные дефекты должны быть задокументированы в системе баг-трекинга. Помимо этого, применяется регрессионное тестирование.
- После завершения дебаггинга предоставляется оценка общего качества продукта.
- После завершения последнего теста процесс тестирования ПО считается законченным.

ПРЕИМУЩЕСТВА ПРОВЕДЕНИЯ ТЕСТОВ НА КАЖДОМ ЭТАПЕ ЖИЗНЕННОГО ЦИКЛА ПО

Пятый этап. Эксплуатация и поддержка

- Даже после достижения стадии релиза продукта, остается необходимость в тестировании, проводимом на этапе эксплуатации и поддержки. Разные пользователи могут работать в абсолютно разных окружениях. Поэтому всегда возможно, что новые ошибки, которые не были выявлены ранее дадут о себе знать. Более того, пользователи могут использовать ПО изначально непредвиденным способом. Это, в свою очередь, может вызвать некоторые непредвиденные проблемы. В таком случае потребуется вмешательство отдела QA.

ПРЕИМУЩЕСТВА ПРОВЕДЕНИЯ ТЕСТОВ НА КАЖДОМ ЭТАПЕ ЖИЗНЕННОГО ЦИКЛА ПО

Заключение

- Очевидно, что процесс управления тестированием ПО затрагивает все этапы жизненного цикла разработки. Он подразумевает сравнение действительного состояния продукта и того состояния, которое было запланировано и задокументировано в плане тестирования продукта. Процесс тестирования, анализа и мониторинга помогает спланировать и изменить последующие задачи наилучшим путем.
- Жизненный цикл тестирования ПО является процессом, которого нельзя избежать. Он непрерывен, продолжителен и требует наличия команды QA, достаточно опытной для того, чтобы произвести полный цикл тестирования. Эта неотъемлемая часть современного процесса разработки ПО помогает заказчику, команде разработчиков, а также конечному пользователю получить продукт высокого качества.
-



СПАСИБО ЗА ВНИМАНИЕ!

ДОМАШНЕЕ ЗАДАНИЕ

ОЗНАКОМИТЬСЯ С
КНИГАМИ

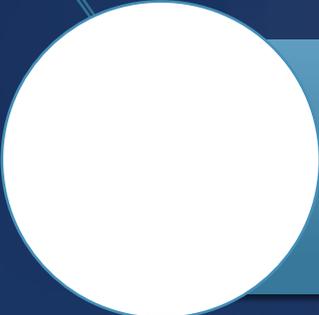
1. КУЛИКОВ СВЯТОСЛАВ
«ТЕСТИРОВАНИЕ ПО.
БАЗОВЫЙ КУРС»
2. САВИН РОМАН
«ТЕСТИРОВАНИЕ
ДОТКОМ»



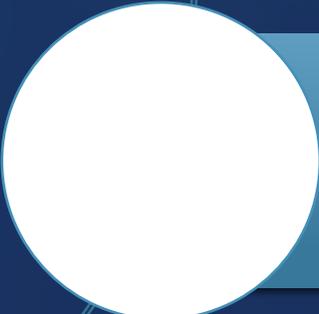
ТЕСТИРОВЩИК
ПРОГРАММНОГО
ОБЕСПЕЧЕНИЯ

КУРС «РУЧНОЕ ТЕСТИРОВАНИЕ»

2. ПРАКТИЧЕСКОЕ ЗАНЯТИЕ



Майндкарты, для чего они нужны и где их создавать



Практика, учимся составлять майнкарты в XMIND

Интеллект-карта, ментальная карта или mindmap — это метод визуализации данных, инструмент, который помогает структурировать информацию и находить взаимосвязи между её частями.

Майндкарта работает так:

1. На карту помещают главный блок, где фиксируется основная идея, концепция, тема или проблема.
2. Вокруг выстраивают другие блоки с идеями, которые связаны с главным блоком.
3. От этих блоков идут ответвления к другим блокам с идеями и так далее.
4. В итоге получается большое дерево решений или ментальная карта.

МАЙНДКАРТЫ, ДЛЯ ЧЕГО ОНИ НУЖНЫ



Интеллект-карта состоит из трёх базовых элементов:

- 1. Ядро или центральный блок** — основной компонент, с которого начинается создание интеллект-карты.
- 2. Ветви** — линии, которые выходят из центрального блока и связывают его с другими блоками, подтемами.
- 3. Остальные блоки** — это ключевые слова, идеи, картинки, образы. Они помогают раскрыть центральный блок, запуская ассоциативные связи, разрабатывающие основную идею из ядра карты.

МАЙНДКАРТЫ, ДЛЯ ЧЕГО ОНИ НУЖНЫ

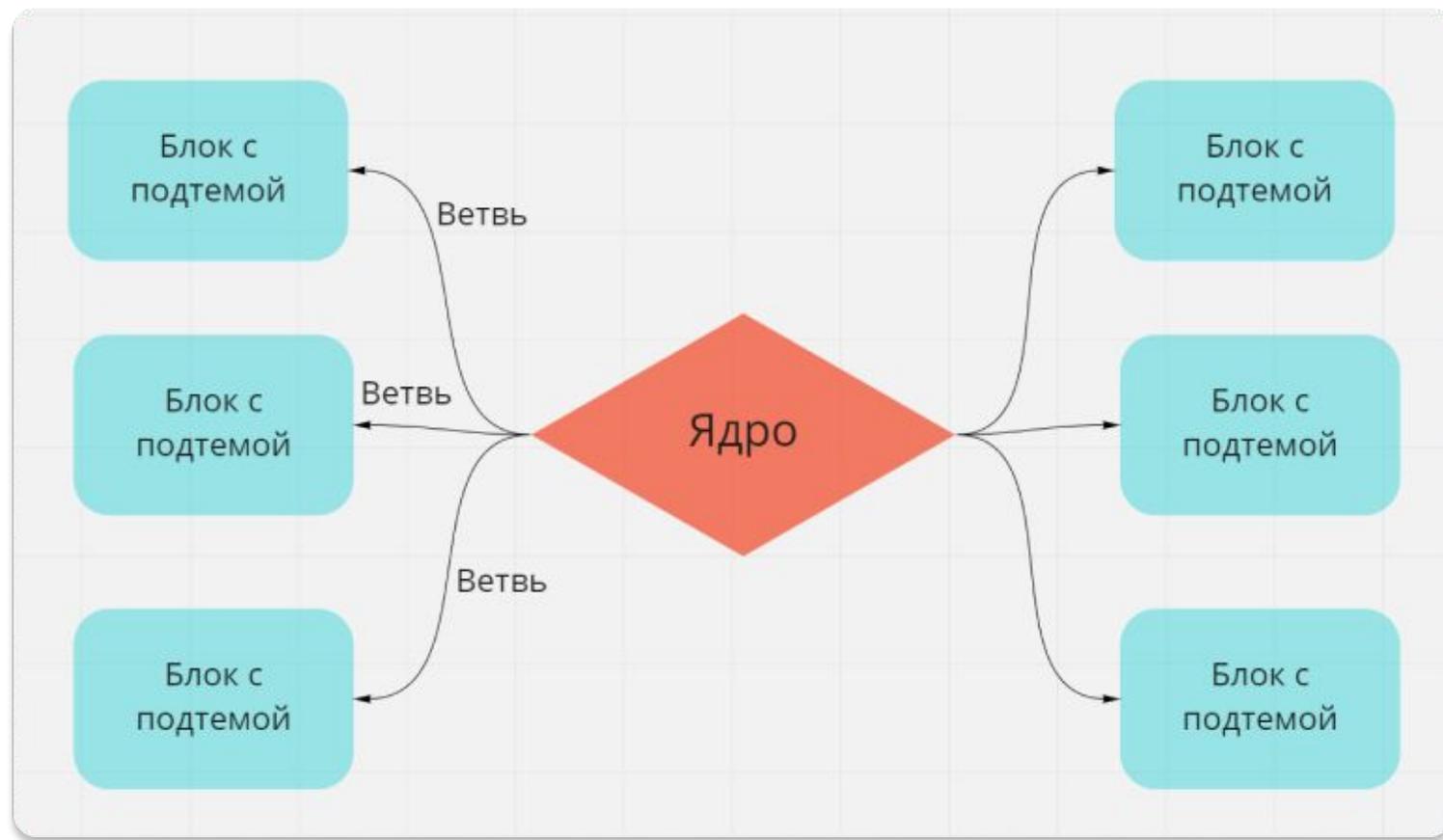


Если в работе вы задаетесь следующими вопросами :

1. С чего начать тестирование?
2. Как ничего не забыть?
3. Как не запутаться в сложном функционале?

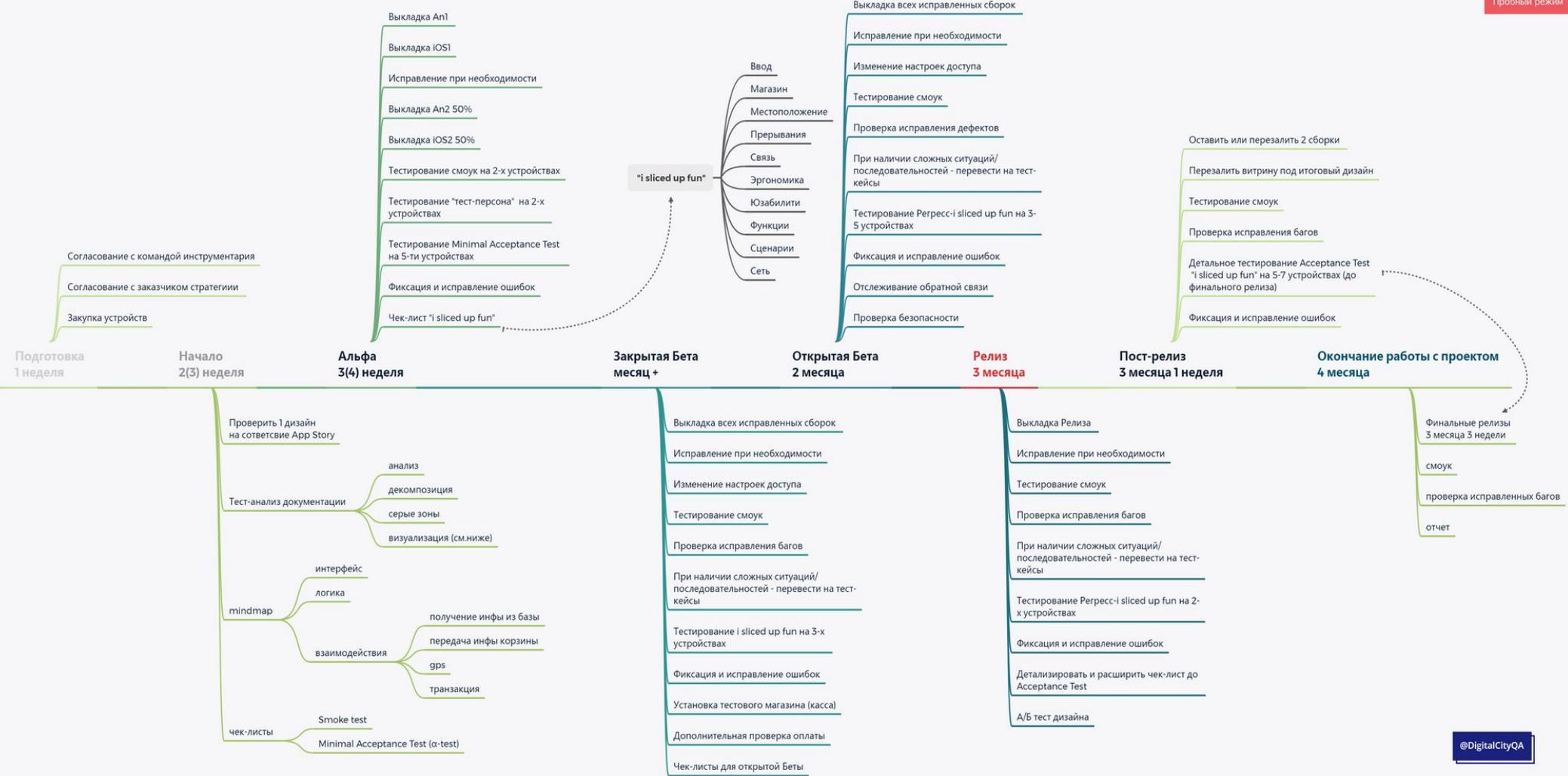
Значит вам, скорей всего, необходимо структурировать ваши мысли и идеи .. А в этом вам поможет MINDMAP.

МАЙНДКАРТЫ, ДЛЯ ЧЕГО ОНИ НУЖНЫ



МАЙНДКАРТЫ, КАК ОНИ ВЫГЛЯДЯТ

Тестирование приложения



ПОСЛЕДОВАТЕЛЬНАЯ МАЙНДКАРТА ТЕСТ-ПЛАНА



MindMeister

Miro

LucidChart

Ayoa

Xmind

ClickUp

Microsoft Visio

Scapple

Google

MindManager

НЕКОТОРЫЕ ПРОГРАММЫ ДЛЯ СОСТАВЛЕНИЯ МАЙНДКАРТ



А ТЕПЕРЬ ПОПРОБУЕМ НА ПРАКТИКЕ СОЗДАТЬ МАЙНДКАРТУ...

