What do we need to test?

* Functionality – features that need to be implemented > FUNCTIONAL TESTING
* How this application will be integrated with bank application? > INTEGRATION TESTING
* How will it work under different browsers and devices? > COMPATIBILITY TESTING
* How will it work on mobile devices during calls and messages? > MOBILE TESTING
* How will it be protected from hackers? > PENETRATION TESTING
* How quickly application will work? > PERFORMANCE TESTING
* How will it be comfortable for people with disabilities? > ACCESSIBILITY TESTING
* What languages should be supported? > LOCALIZATION TASTING
* How clear and comfortable will this application be for users? > USABILITY TESTING

**TEST STRATEGY**

Clarify:

* What do we need to test?
* What are timelines?

Define test strategy:

* What testing types will be in scope (scope – масштаб).
* How they will be performed, combined and optimized.
* How testing will be organized into the phases or levels.

Тестовая стратегия включает в себя три основных вопроса:

* Какие тестовые типы у нас будут в рамках данного тестирования.
* Как мы будем выполнять это тестирование. Как мы можем объединить или оптимизировать тестирование.
* Что именно будет выполнять в рамках определенных фаз.

How to organize testing?

Some testing types can be combined:

* Regression & Localization & Mobile & Compatibility vs Permissions

Some testing types require person with special skills or /and budget allocation:

* Performance, security, usability

Some testing types require “special testing”/business knowledge:

* Accessibility, Localization, Mobile testing.

When testing types will be performed?

* During sprint/at the end of release/during UAT; one – time execution or every sprint.

*Регрессионное тестирование* – это когда необходимо проводить проверку того, что тестировали раньше.

*Локализационное тестирование* — это проверка содержимого приложения или сайта на соответствие лингвистическим, культурным требованиям, а также специфике конкретной страны или региона. *Тестирование локализации* — один из видов контроля качества, который проводится во время разработки продукта.

TESTING APPROACH (подход тестирования):

Testing environments:

Who provided env? What requirements (*Какие требования*)? Who supports them? When are they refreshed?

Test Tools:

Test Management, Bug Tracking tools, mockups, scripts, special like Jmeter, etc.

Testing Phases:

When and who perform particular testing types?

Test Data Management:

Production like data? Specific data? Large amount of data for performance?

Test Documentation:

Test Cases or just checklist or Test Scenarios or Test Matrix?

Test Automation:

Who defines what to automate? Based on what criteria? How will it help to manual testing? When do we run what?

Time is crucial factor for success.

Main limitation to speed development is quality.

Необходимо найти баланс между: Money – Time – Quality.

**Testing in Agile.**

Engineering practices:

CI/CD

Automated Quality Gates

Coding Standards

Code review

Automated static code analysis

Unit testing

Test-Driven Development

Release strategy

TEST AUTOMATION IN SPRINT

1. BEFORE DEVELOPMENT:

* Collaborate with developer to define (определить) what tests will be created on which level and by whom.
* Agree with developer about unique ID of controls (request/response templates in case of services implementation).

1. DURING DEVELOPMENT:

* Create empty tests with idea of the tests.
* Populate tests with steps: include available steps and comment out steps which are dependent on implementation.
* Mock components which will not be available during this sprint.
* Automate test data generation/population.

1. ONCE CODE IS READY:

* Debug new automation tests as soon as code is committed.
* Run all existing automation tests after commit of new code.

How to improve test automation?

Test automation issues:

Do not find defects.

Do not reduce manual test effort.

Tests are unstable.

Execution is too long.

Time-consuming maintenance.

Feedback on product quality is provide late

How to resolve:

Align test automation with common test strategy.

Balanced test automation pyramid to cover more business logic on API level.

Include test automation (at least smoke test) into CI/CD pipeline and quality gates.

Parallel execution.

Run and analyze result at least once in sprint.

REQUIREMENT TESTING. INTRODUCTION.

Product backlog разделяют на user stories. Составляющие user story: Requirements; Acceptance criteria (критерии приемлемости); Priority. Requirements: Description of what need to implemented, system behavior. Acceptance criteria: Set of conditions required to validate user story. Priority: Importance of user story implementation.

Properties (свойства).

User story: independent (независимый), negotiable (подлежащий обсуждению), valuable (ценный / полезный), estimable (достойный оценки), small, testable.

Acceptance criteria: specific, measureable, assignable (присваиваемый), realistic, time-related (связанные со временем).

Requirements: complete (полный), correct, feasible (осуществимый), consistent, unambiguous (недвусмысленный).

Что же делать, когда у нас есть вопросы? Сейчас у нас есть (в идеале) всегда доступный бизнес-аналитик (либо продукт-овнер), которому можно задавать вопросы. Цена ошибки в любом требовании достаточно высока. Поэтому идеально начать работать с требованиями до начала разработки. Есть такое понятие, как refinement meeting (его ещё называют груминг митингом). Он должен происходить даже до начала планирования самого спринта и тем более разработки. В идеале, продукт-овнер заранее предоставляет нам требование, acceptance criteria, описание, кот. мы должны заранее проанализировать. Во время митинга мы общаемся с продукт-овнером, задаем вопросы, что нам не понятно.

Test scenarios (for exemple):

Positive:

Verify money transfer to cardholder’s own card with amount > 100 USD.

Verify money transfer to not cardholder’s card with amount < 100 USD.

Verify money transfer to not cardholder’s card with amount > 100 USD.

Negative:

Verify money transfer with amount = 0 USD.

Verify money transfer from the card issued not in USD.

Boundary:

Verify money transfer to not cardholder’s card with amount = 100 USD.

Verify money transfer with amount < card amount + fee.

Verify money transfer with amount > card amount , but < card amount + fee.

Interaction with other application functionality:

Verify money transfer from not active card.

Verify money transfer to invalid card.

…

Виды тестирования.

По объекту тестирования:

1. Функциональное:

функциональное тестирование

тестирование GUI

тестирование безопасности

1. Нефункциональное:

интерфейсов

юзабилити

локализации

конфигурационное

совместимости

инсталяционное

производительности

помехоустойчивости

документации

По степени автоматизации:

1. Ручное.
2. Автоматизированное.

По критерию изменений:

1. Smoke тесты (выполняются ли основные функции)
2. Регрессионное тестирование
3. Тестирование билда
4. Санитарное тестирование

Тестирование по уровням:

1. Модульное тестирование
2. Интеграционное тестирование
3. Системное тестирование
4. Уровень критического пути