

РАЗРАБОТКА ПО

Author:

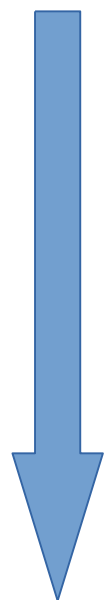
Содержание

1. Процесс разработки ПО
2. CI/CD
3. Тестирование
4. Инструментарий
5. Git

SOFTWARE DEVELOPMENT PROCESS

- ♦ Цели — решение бизнес задач
- ♦ Методы — подходы к организации процессов и решению задач
- ♦ Средства — ресурсы:
 - Время
 - Деньги
 - Люди
 - «Железо»
 - ПО
- ♦ Экспертиза — уровень владения знаниями и навыками, а также оценка всего вышеперечисленного на всех этапах

Как это работает и кто все эти люди



- Идея
- Анализ
- Формулирование требований
- Проектирование
- Разработка
- Тестирование
- Внедрение
- Техническая поддержка

- Customer
- Project Owner
- Project Manager
- Business Analyst
- Architect
- Developers
- Testers(QA)
- DevOps

- Stakeholders
- Team Lead

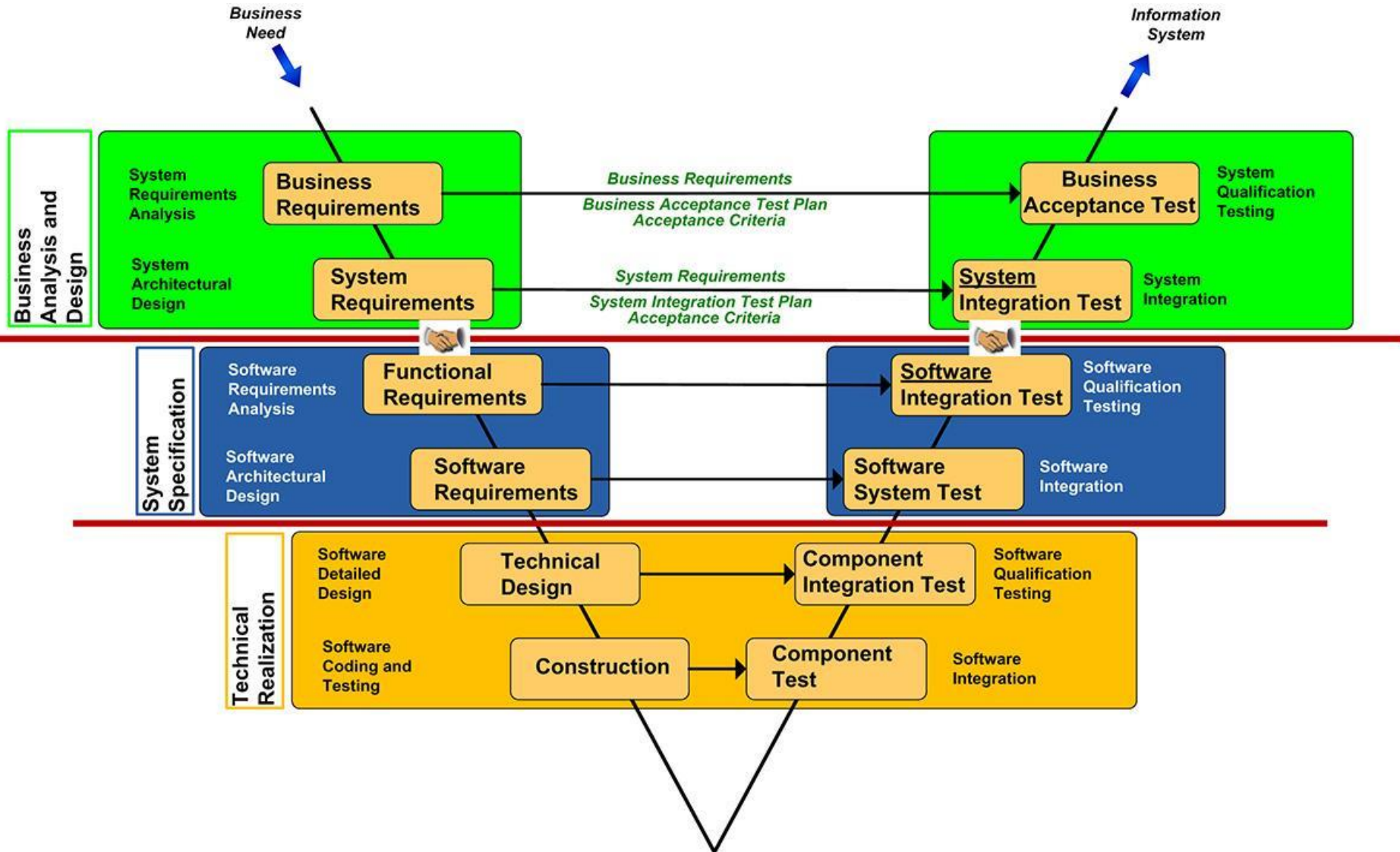
Business Analyst



Project Manager



V - model



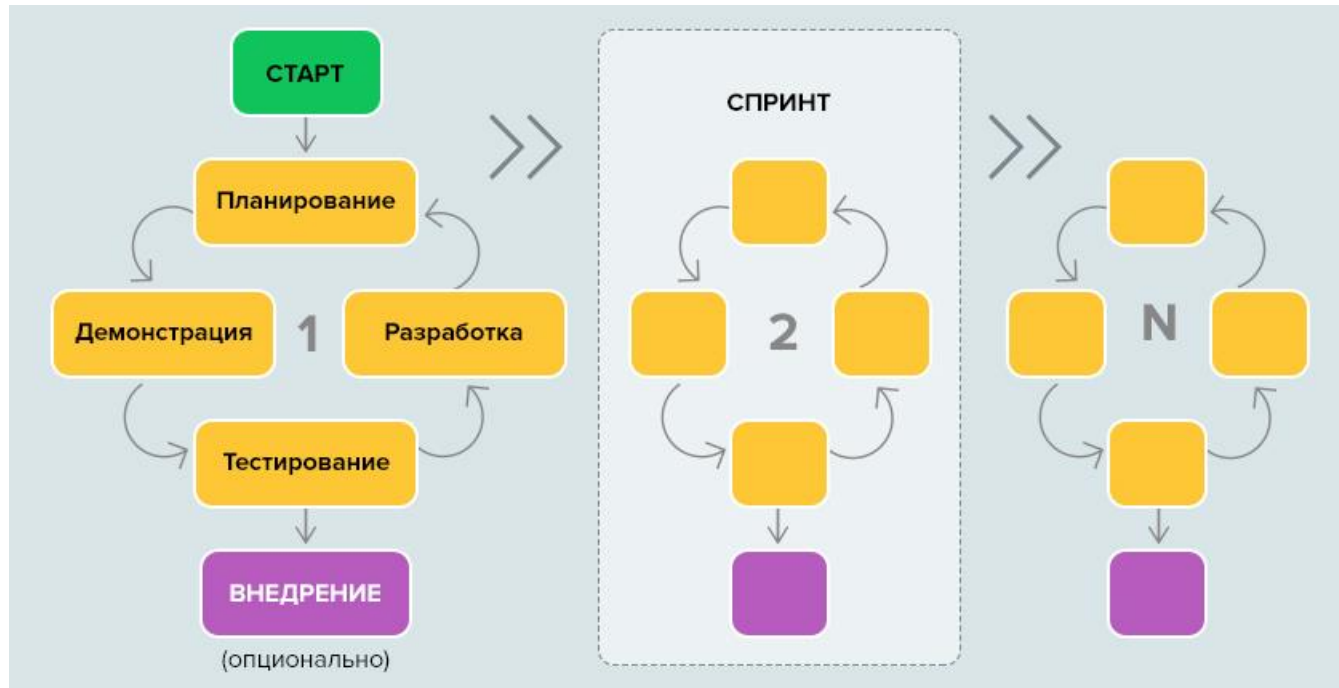
Waterfall



Incremental model



Agile



- люди и взаимодействие важнее процессов и инструментов
- работающий продукт важнее исчерпывающей документации
- сотрудничество с заказчиком важнее согласования условий контракта
- готовность к изменениям важнее следования первоначальному плану

Взаимодействие в команде

У нас была 2pizza team, Jira и SCRUM мастер...

Jira Interface Overview:

- Navigation:** Dashboards, Projects, Issues, Boards, Create.
- Left Sidebar:** Teams in Space, Backlog, Active sprints, Releases, Reports, Issues, Components, PROJECT SHORTCUTS, TIS Roadmap, TIS Team Org Structure, Give feedback, Project administration.
- Main Board:** All sprints. QUICK FILTERS: Product, UI, Server, Only My Issues, Recently Updated.
- Columns:** 12 To Do, 4 In Progress, 1 Code Review, 7 Done.
- Issues:** TIS-37, TIS-10, TIS-8, TIS-68, TIS-17, TIS-26, TIS-12, TIS-33, TIS-15, TIS-20, TIS-67, TIS-56, TIS-45, TIS-49, TIS-69, TIS-23.

Issue TIS-67 Details:

- Summary:** Developer Toolbox does not display by default
- Attachments:** Screen Shot 2015-08-13 at 4:1 326 kB 20/Aug/15 12:08 PM
- Sub-Tasks:**

Issue Key	Summary	Status	Actions
TIS-127	Check Java version	OPEN	[Edit] [Close]

- Development:**

Item	Updated
1 branch	Updated 17/May/14 7:32 AM
7 commits	Latest 17/May/14 7:30 AM
1 pull request	Updated 17/May/14 7:32 AM
3 builds	Latest 16/May/14 2:31 PM

- Deployed to Staging and Production**

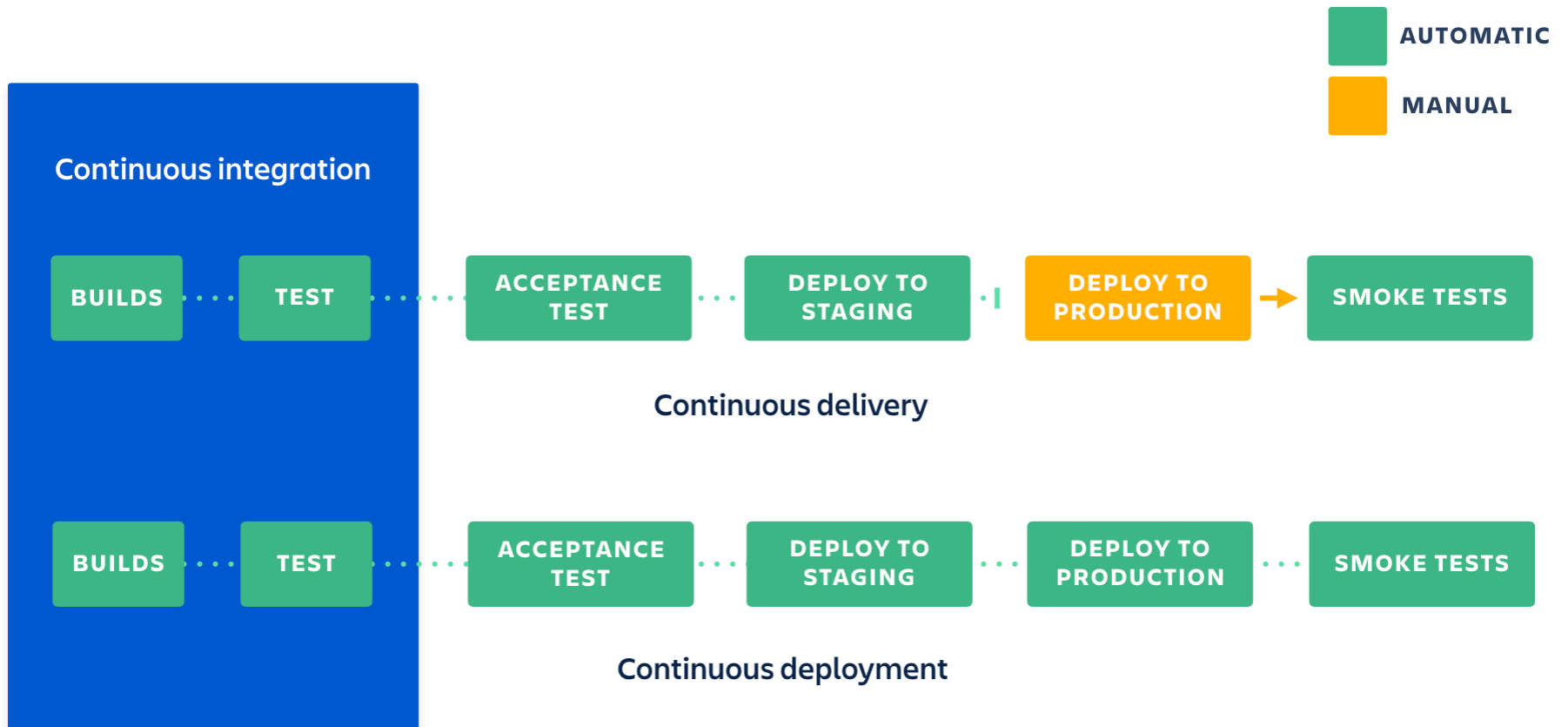
CONTINUOUS INTEGRATION DEPLOYMENT DELIVERY

CI/CD — долой рутину

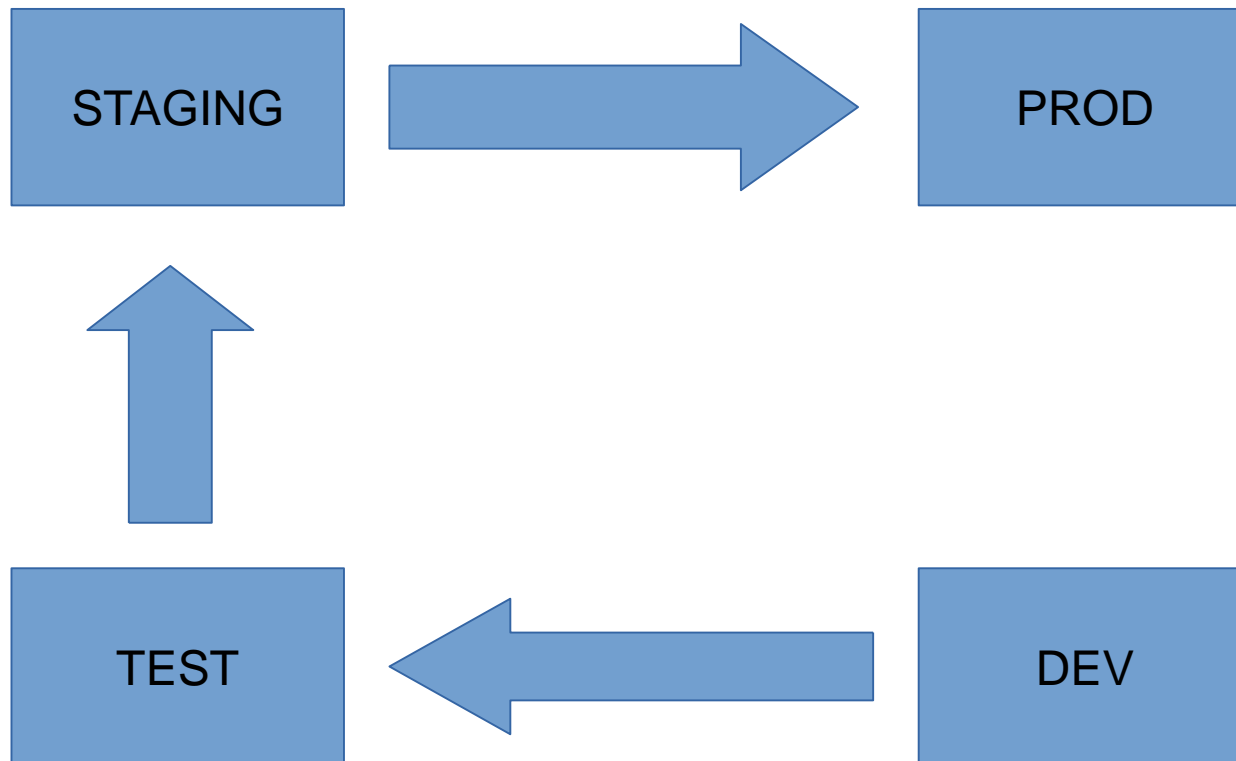
Цели:

- Организация параллельной работы
- Сокращение времени разработки и доставки программного продукта
- Выявление багов на ранних этапах
- Освобождение от рутинных операций

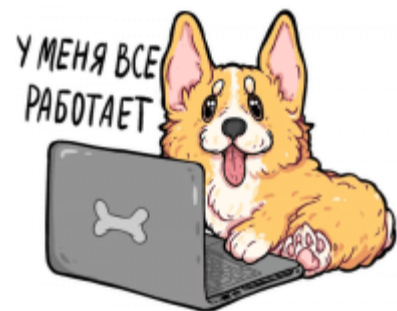
Pipelines



Среды/стенды/environments



SOFTWARE TESTING (QA)

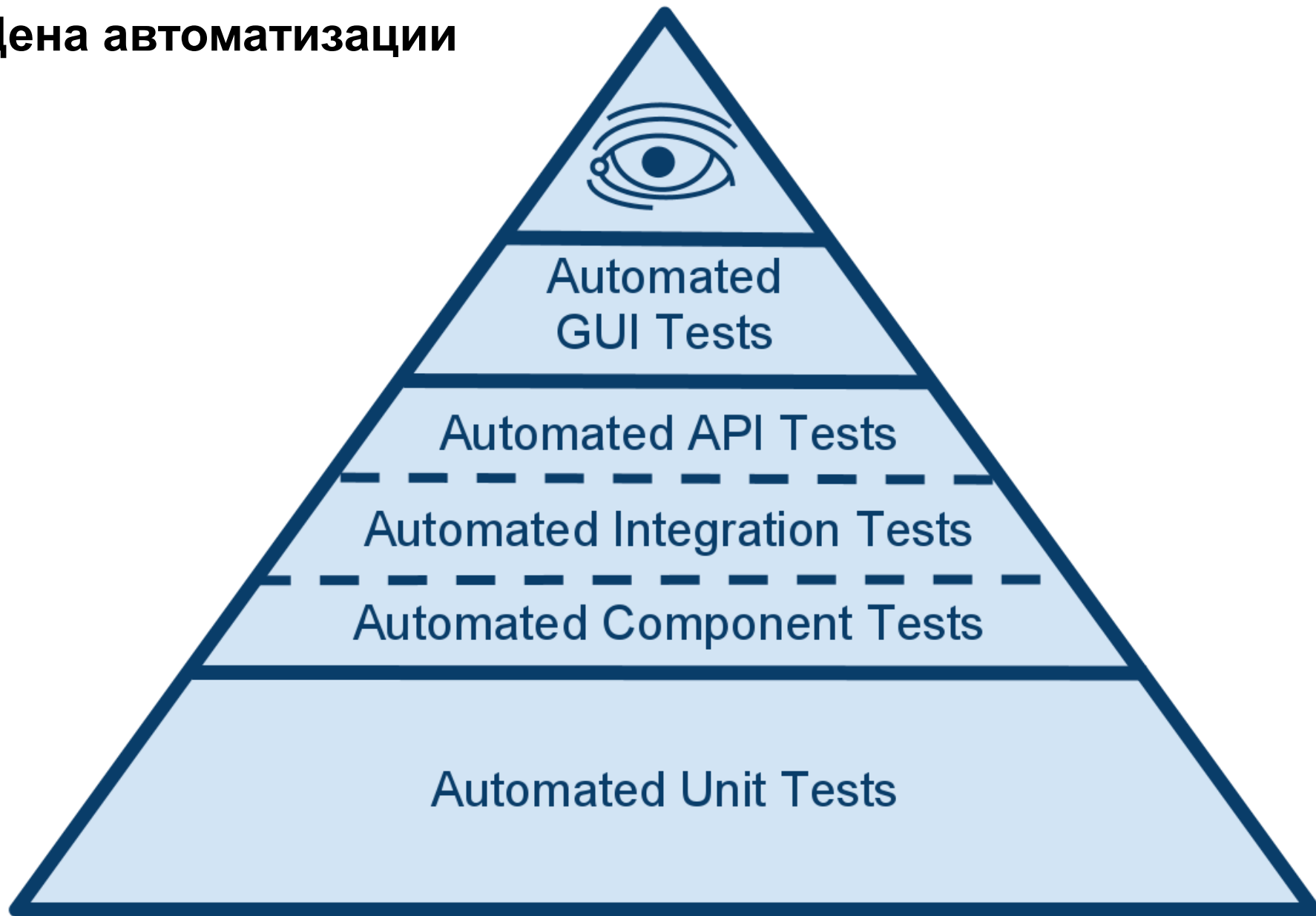


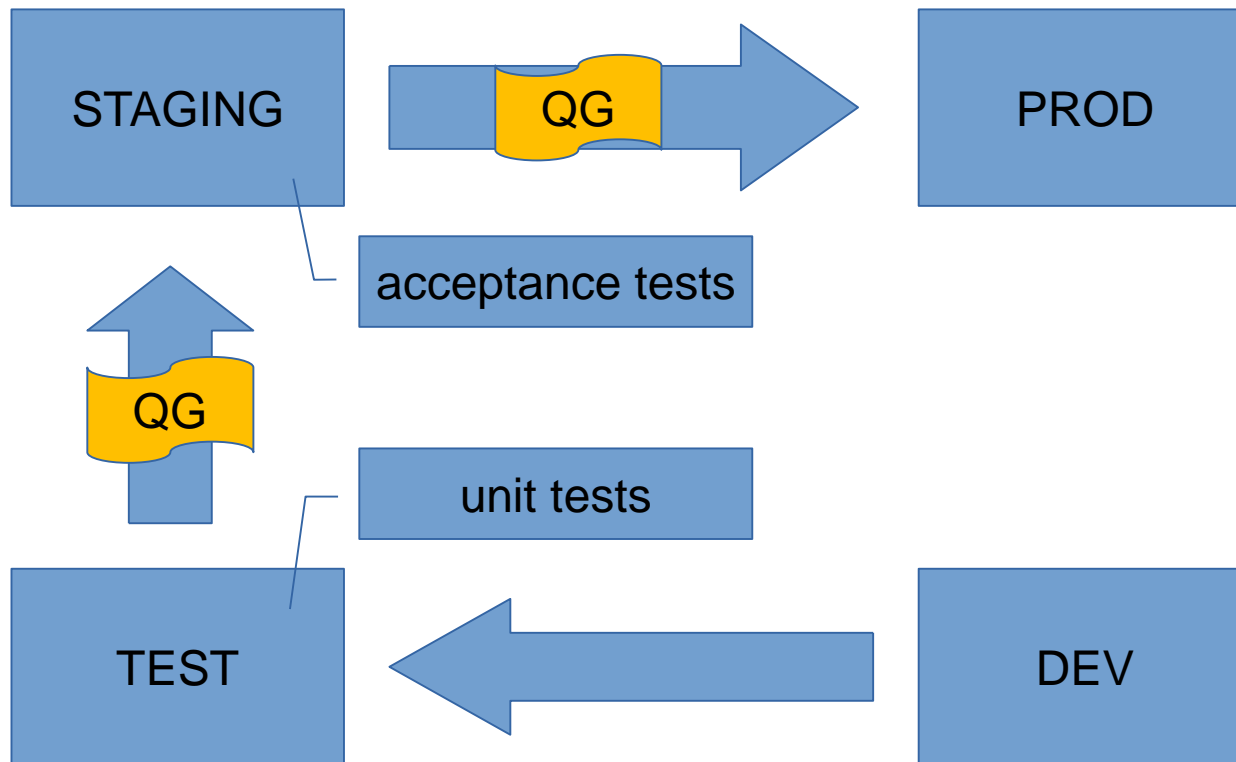
Виды тестов

- **функциональное тестирование (Functional testing)**
 - ◊ unit testing
 - ◊ API testing
 - ◊ GUI testing
- **нефункциональное тестирование (Non-functional testing)**
 - ◊ usability testing
 - ◊ load testing
 - ◊ security testing
 - ◊ stress testing
- **структурное тестирование (Structural testing)**
 - ◊ тестирование логики (white/black box)
- **тестирование изменений (Change related testing)**
 - ◊ regression testing
 - ◊ retesting

<https://training.qatestlab.com/blog/technical-articles/review-the-types-of-testing/>

Цена автоматизации





SOFTWARE DEVELOPMENT TOOLS

Integrated Development Environment

MS Visual Studio

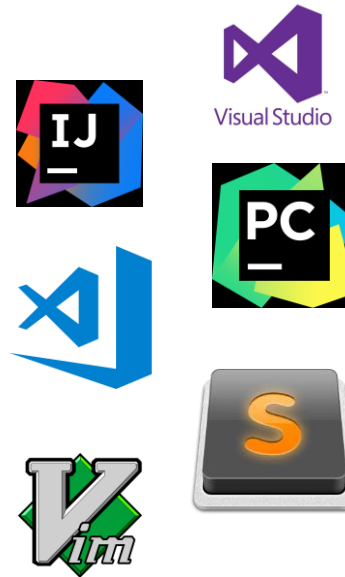
IntelliJ Idea

PyCharm

Visual Studio Code

Sublime Text

VIM



Языки



Классификация языков по логике сборки для девопсов



Компилируемые в бинарники:

C, C++, Go, Pascal, Objective-C, Haskell, D, Rust



Компилируемые в байткод:

Java, C#, Python, Swift, Kotlin, Groovy, F#



Интерпретируемые:

Python, Ruby, JavaScript, PHP, PowerShell, R, VBScript, PostScript



Version Control System

DVCS

Git

Mercurial

SVN



TortoiseSVN

Source Code Managment

Code Review

GitLab



Gerrit



Bitbucket



GitHub



Microsoft Team Foundation
Server



Хранение артефактов

Фаловая система
Облако
Sonatype Nexus
Artifactory



Результатом сборки могут быть:

- бинарник (exe, ELF)
- пакет (rpm, deb, wheel, jar, apk)
- архив
- образ (docker, iso, wim)

Системы сборки и контроля зависимостей

Java: Maven, Gradle, Ant



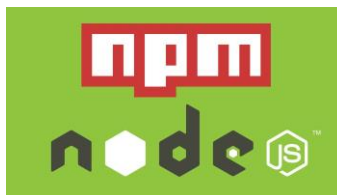
Python: PIP



C#: .Net build



JavaScript: npm



CI/CD

Jenkins



TeamCity



GitLab CI



Drone



Тесты

JUnit



Unittest



Selenium



SonarQube



GIT

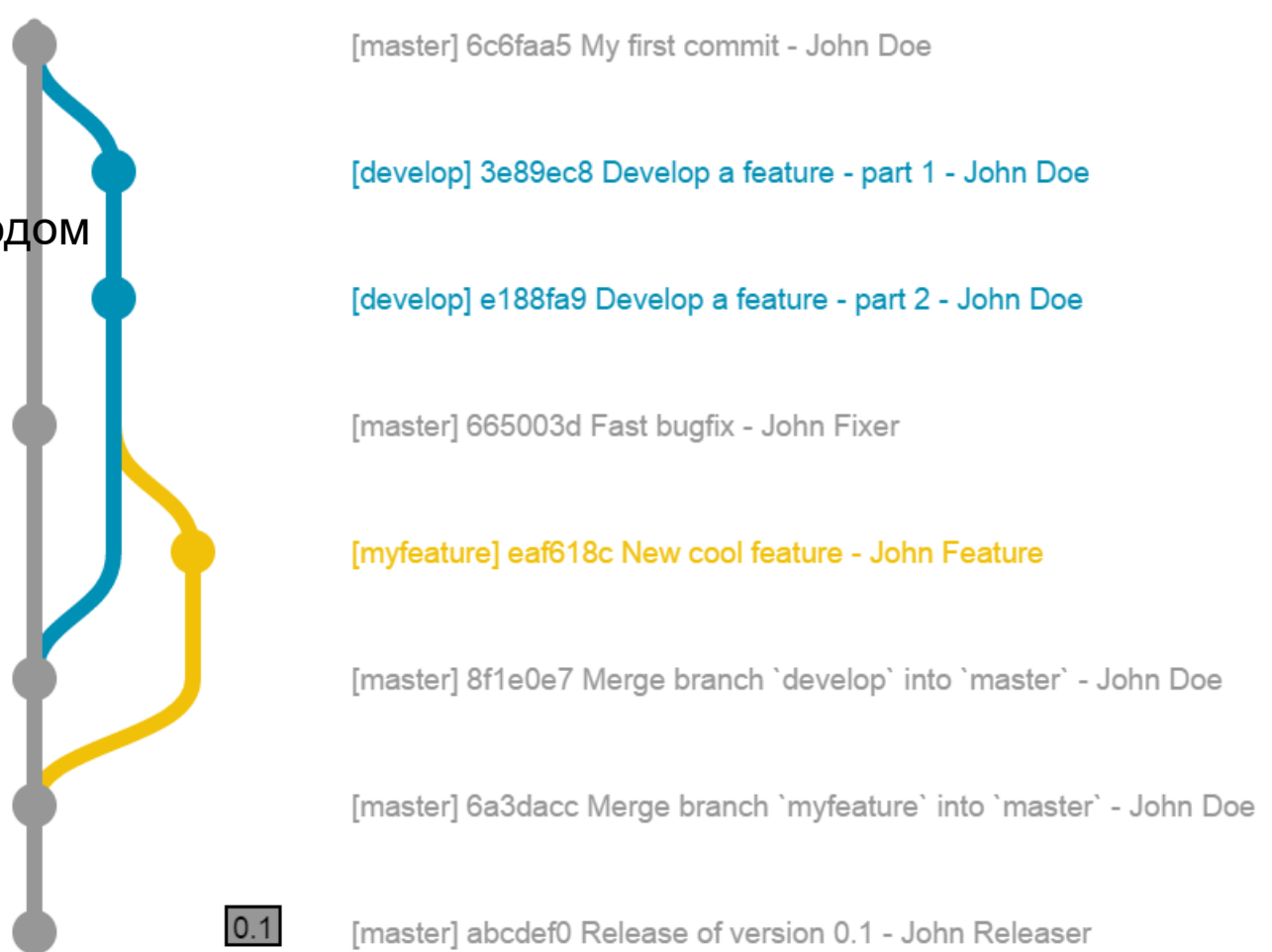


Версионирование

Параллельная работа с кодом

Разрешение конфликтов

Ветвление кода



Что в гит запущено — назад не воротишь

В Git'е храним только код — никаких бинарников, архивов, образов!

Основные операции

`git clone <repo>`

`git checkout my-branch`

`git add ./*`

`git status`

`git commit -m "new cool feature"`

`git log`

`git push origin`

`git checkout master`

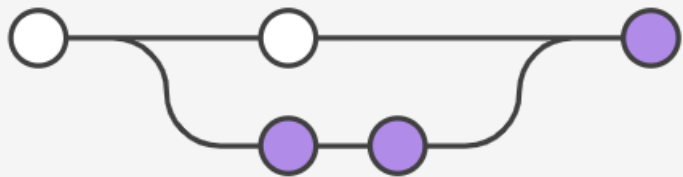
`git merge`

`git push origin`

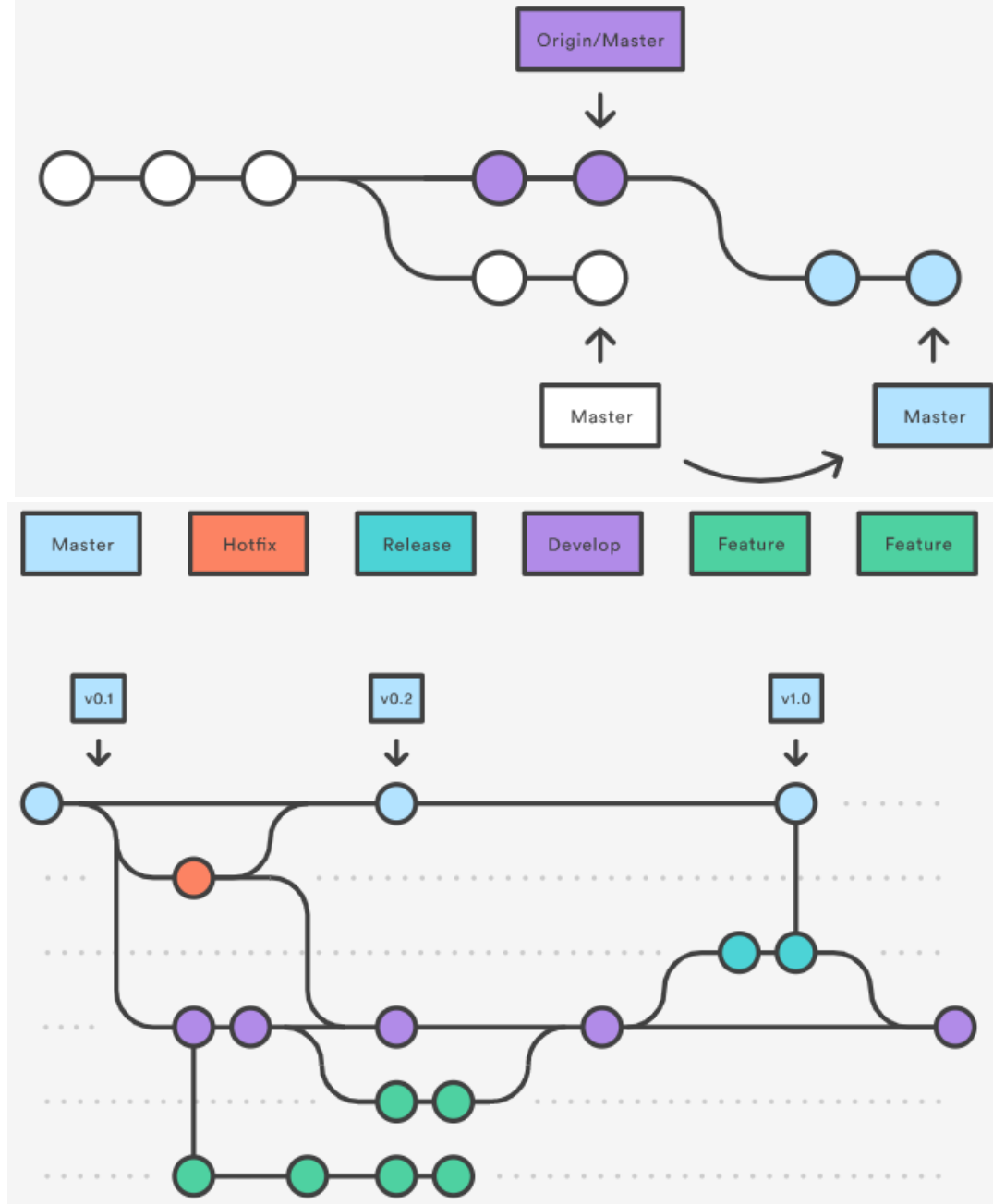
Branching strategies

Centralized (rebase) →

Feature Branch Workflow



GitFlow →



Домашнее задание

Познакомиться с методологиями подробнее.

<https://training.qatestlab.com/blog/technical-articles/popular-software-development-life-cycles/>

Прогуглить весь инструментарий.

Завести гит (GitHub, ButBucket), скачать cheat sheet, настроить клиент (имя/почта), потренироваться — <https://try.github.io/>

Thanks for Your Attention

Questions?

Author: