SQA Basics

Janna Spektor
Portnov Online

What is Quality?

2 perspectives:

- Customer’s satisfaction;
- Product being under requirements;

BUT…

1) Customer satisfaction is a subjective term.
   - It will depend on who the “customer” is.
   - Each type of customer will have their own view on “quality”.

2) Meeting the requirements doesn’t guarantee customer’s popularity.

What is Software Quality?

Software Quality *is how close the actual software product is to the expected (intended) product*.

Quality Software:

- reasonably bug-free;
- delivered on time and within budget;
- meets requirements and/or expectations;
- maintainable.

What is Software Quality Assurance?

Software Quality Assurance is the process of monitoring and improving *all activities associated with software development*, from requirements gathering to coding, testing and implementation.

It involves:

- Monitoring and improving the process;
- Following agreed-upon standards and procedures;
- Detecting and dealing with problems at the earliest possible stage

What is Software Testing?

Software testing is a process used to identify the correctness, completeness, and quality of developed computer software.

What is Software Testing?

It includes a set of activities conducted with the intent of finding errors in software so that it could be corrected before the product is released to the end users:

- analyzing the software;

- detecting differences between existing and required conditions;

- evaluating the features of the software.

Purpose of Software Testing

- **Verification**

  Are we building the system right?

  Check the actual results against the requirements;

- **Validation**

  Are we building the right system?

  Check that our system is what the user actually wanted;

- As a result - **error detection** (in order to find and fix the problems)

  Find if things happen when they shouldn't or things don't happen when they should.

SQA vs Testing

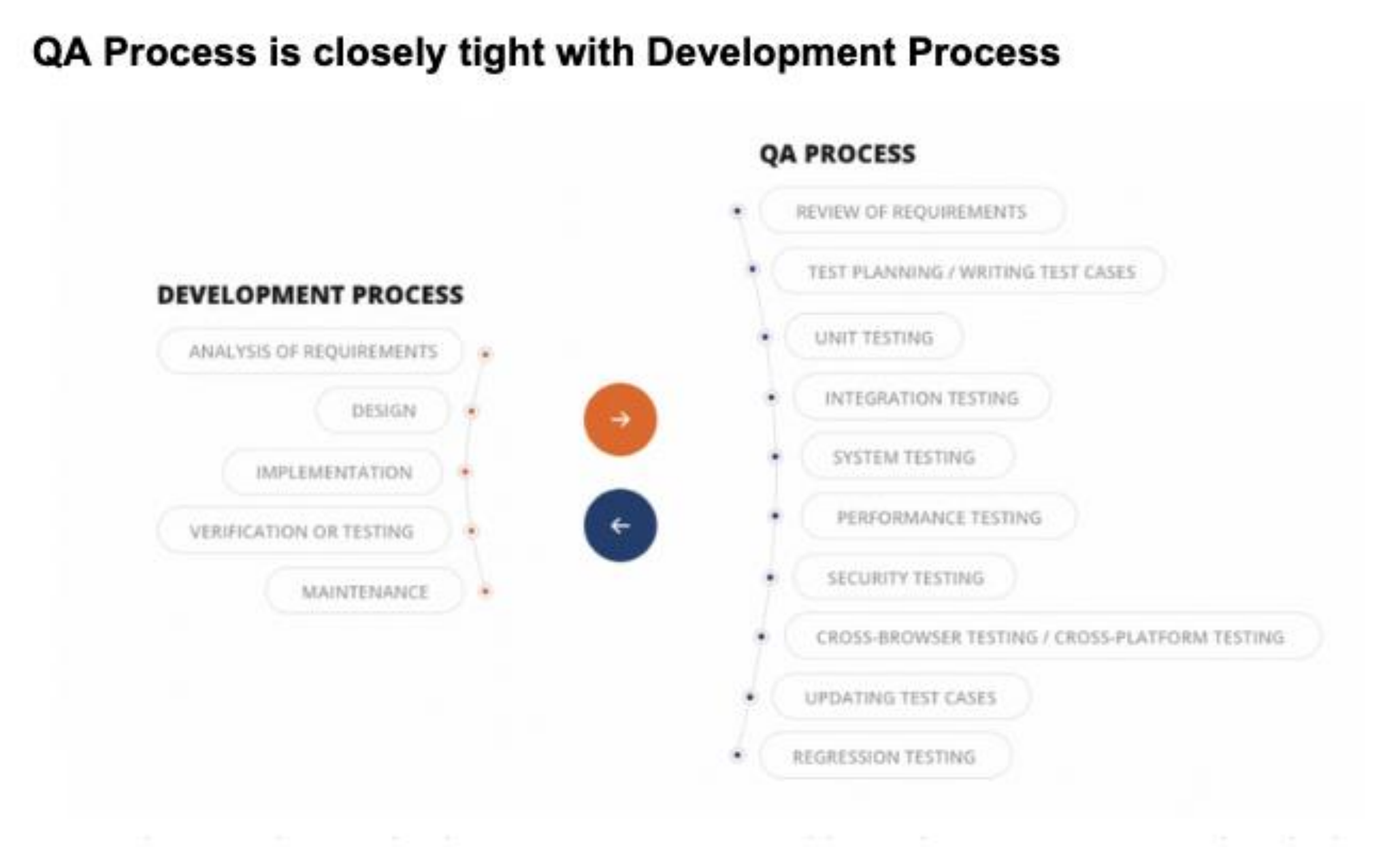
Software Quality Assurance is “preventative” and about the whole process of software development, covers all phases of Software Development.

Software Testing is mainly about testing/verifying the software to detect the differences between existing and required conditions (requirements)

Software Testing Life Cycle (STLC) Phases are:

- Requirements Analysis
- Test Planning
- Test Development
- Test Execution
- Defect Management
- Test Reporting

Testing Process:



QA Process is closely tight with Development Process

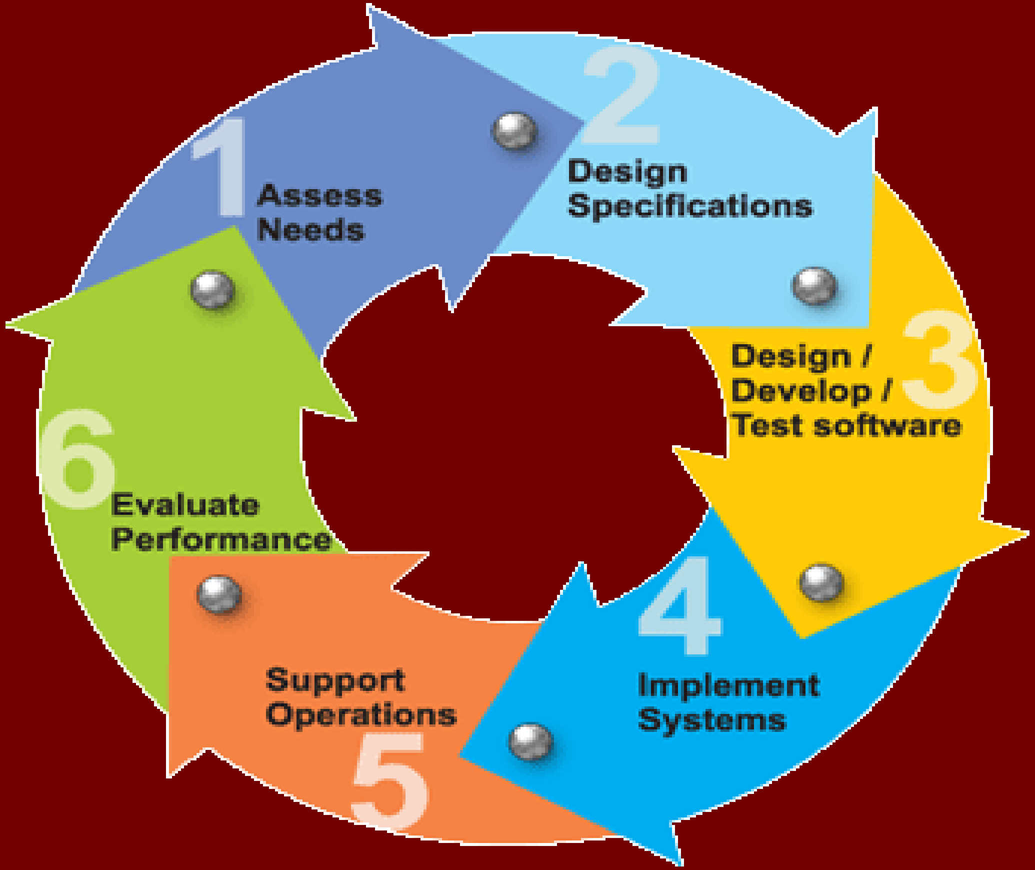
DEVELOPMENT PROCESS

- ANALYSIS OF REQUIREMENTS
- DESIGN
- IMPLEMENTATION
- VERIFICATION OR TESTING
- MAINTENANCE

QA PROCESS

- REVIEW OF REQUIREMENTS
- TEST PLANNING / WRITING TEST CASES
- UNIT TESTING
- INTEGRATION TESTING
- SYSTEM TESTING
- PERFORMANCE TESTING
- SECURITY TESTING
- CROSS-BROWSER TESTING / CROSS-PLATFORM TESTING
- UPDATING TEST CASES
- REGRESSION TESTING

SDLC



1 Assess Needs

2 Design Specifications

3 Design / Develop / Test software

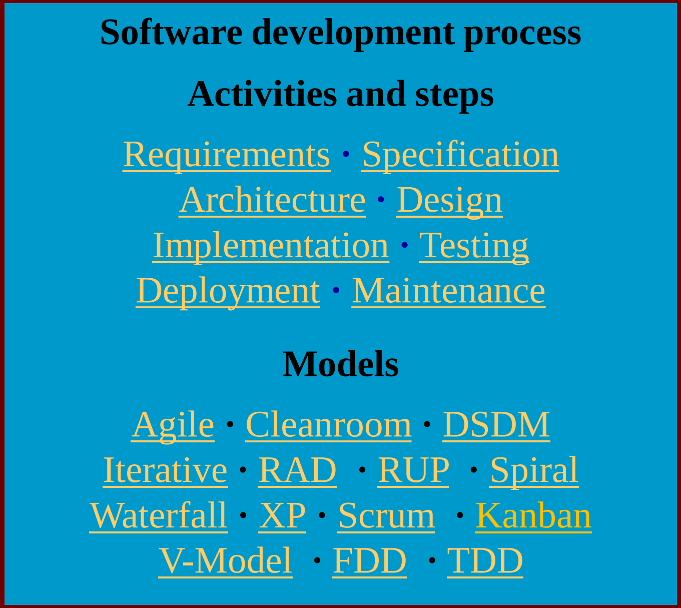
4 Implement Systems

5 Support Operations

6 Evaluate Performance

Software Development Life Cycle (SDLC)

Software Development Life Cycle (SDLC) is a conceptual model used in project management to describe the stages involved in development of an application, from an initial feasibility analysis through maintenance of the completed application.



Software development process

Activities and steps

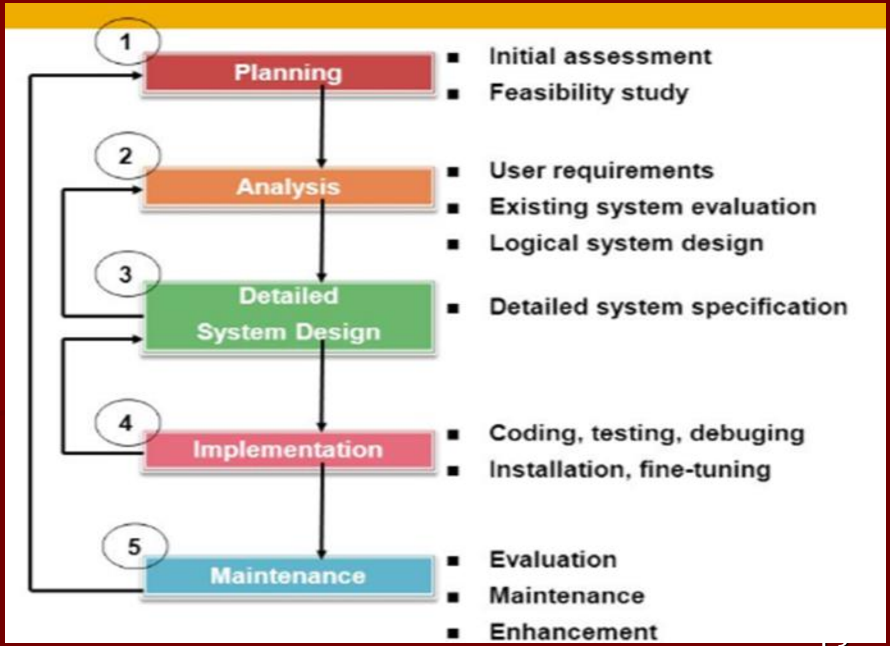
Requirements · Specification
Architecture · Design
Implementation · Testing
Deployment · Maintenance

Models

Agile · Cleanroom · DSDM
Iterative · RAD · RUP · Spiral
Waterfall · XP · Scrum · Kanban
V-Model · FDD · TDD

SDLC - Phases (Example)

- **Planning**
  - Concept development & Planning
- **Analysis**
  - Requirements gathering & Analysis
- **Design**
  - Architecture & Specifications
- **Development**
  - Development
  - Test
  - Implementation
- **Maintenance**



1 Planning
- Initial assessment
- Feasibility study

2 Analysis
- User requirements
- Existing system evaluation
- Logical system design

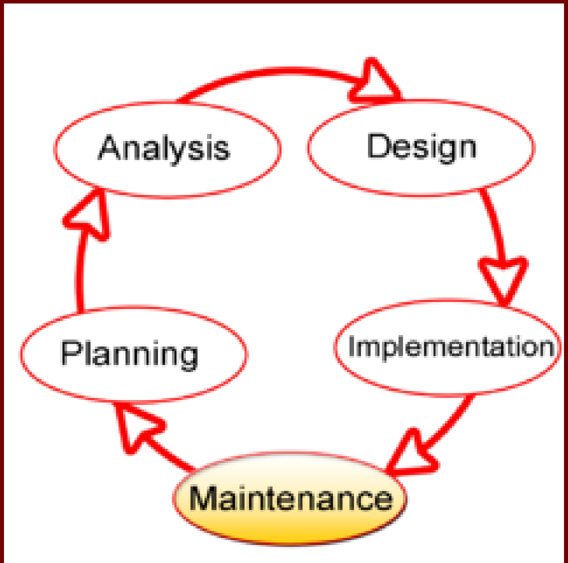
3 Detailed System Design
- Detailed system specification

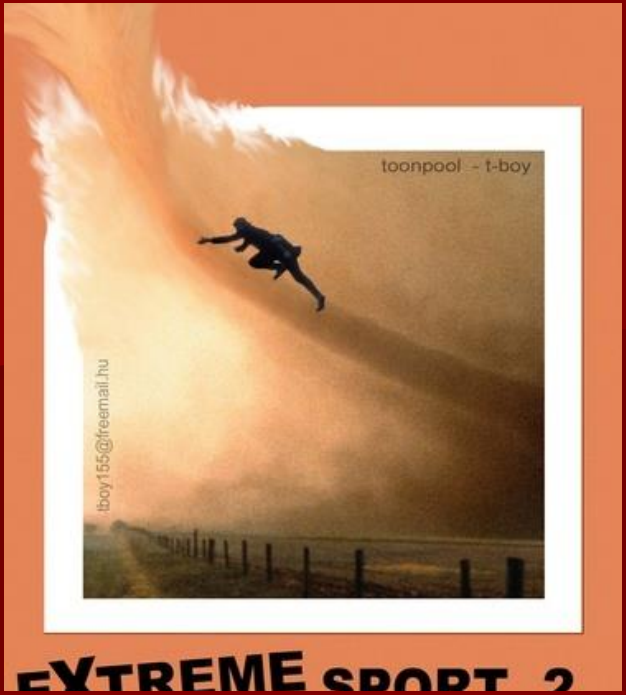
4 Implementation
- Coding, testing, debuging
- Installation, fine-tuning

5 Maintenance
- Evaluation
- Maintenance
- Enhancement

SDLC – Participants (example)

- Product Manager (Business Analyst);
- Technical Writer;
- System Architect;
- Project Manager;
- DBA;
- Networking;
- Developers/Software Engineer;
- **SQA Engineer;**
- Build/Release Engineer;
- Performance Engineer;
- Security/Backup Engineer;
- Technical Support





Development Models

Software development process

Activities and steps

Requirements · Specification
Architecture · Design
Implementation · Testing
Deployment · Maintenance

Models

Agile · Cleanrom · DSDM
Iterative · RAD · RUP · Spiral
Waterfall · XP · Scrum · Lean
V-Model · FDD · TDD

Development Models

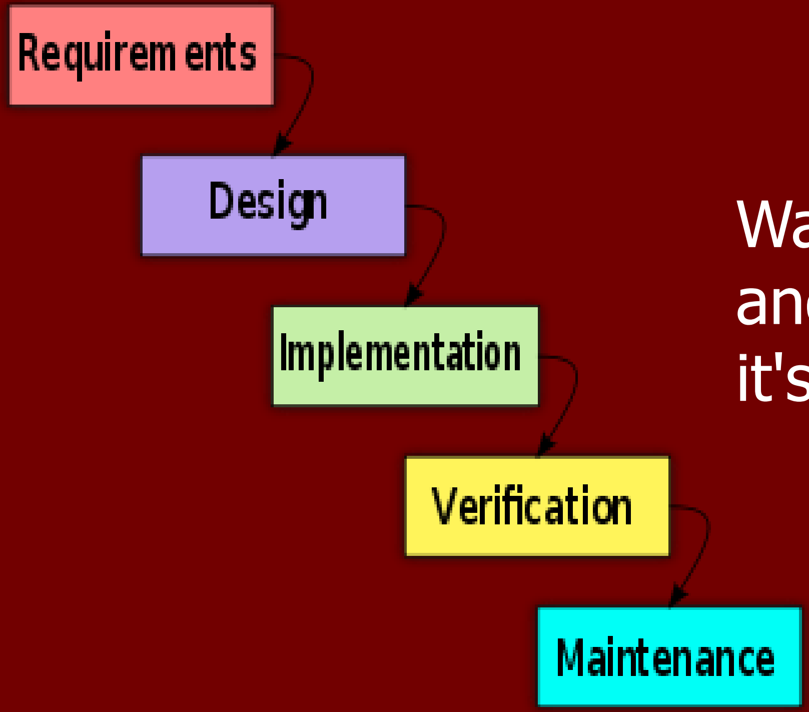
- Provides basis for project planning, estimating & scheduling
- Provides standard set of terminologies, activities & deliverables
- Provides mechanism for project tracking & control
- Increases visibility of project progress to all stakeholders

Development Model

- **Some examples of popular SDLC are:**
- **Waterfall** (Traditional-sequential)
- **Agile**/Extreme: Scrum, XP (extreme programming), Kanban
- And more…

Waterfall Model

Waterfall model is a sequential software development process, in which progress is seen as flowing steadily downwards (like a waterfall) through the phases of Conception, Initiation, Analysis, Design (validation), Construction, Testing and Maintenance.



Waterfall discourages revisiting and revising any prior phase once it's complete.

Waterfall Model Strengths

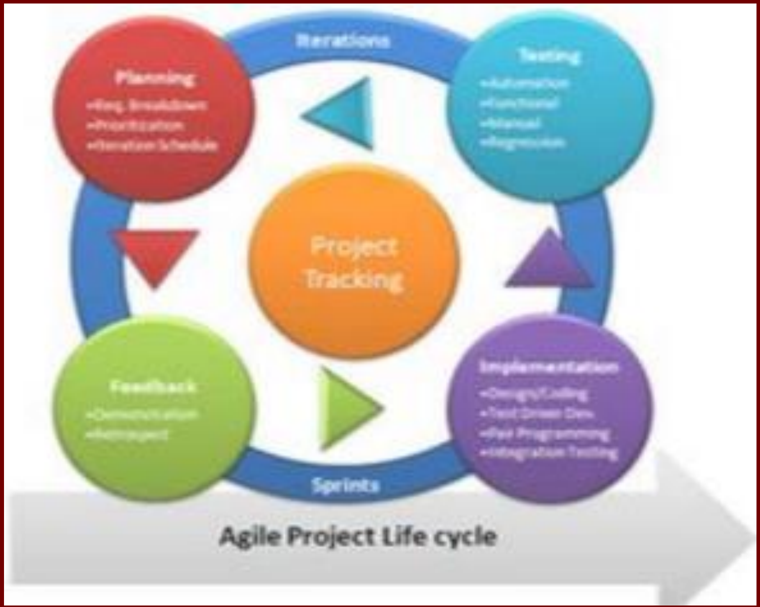
- Easy to understand and use: *phases are processed and completed one at a time*
- Provides structure to the project: each phase has specific deliverables and a review process
- Milestones and requirements are well understood and defined
- Sets requirements stability
- Good for management control: plan, staff, track, schedule, deadlines and so on

Waterfall Model Weaknesses

- All requirements must be fully specified upfront
- Not suitable for the projects where requirements are at a moderate to high risk of changing
- Deliverables created for each phase are considered frozen – inhibits flexibility
- Difficult to go back and change something that was not well-thought out in the concept stage
- Not a good model for complex and object-oriented projects.
- Little opportunity for customer to preview the system

Agile Model

- Agile is a software development model that promotes incremental development and collaboration between self-organizing, cross-functional teams.



Manifesto for Agile Software Development

The agile emphasizes on four core values:

- Individual and team interactions over processes and tools
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following a plan

Scrum Team

- It consists of three roles:
- Scrum Master - responsible for setting up the team, sprint meeting and removes obstacles to progress
- Product owner - creates product backlog, prioritizes the backlog and is responsible for the delivery of the functionality at each iteration
- Developers- Team manages its own work and organizes the work to complete the sprint or cycle

Agile: Scrum and Kanban

- **Scrum** teams commit to ship working software through set intervals called sprints.
- Scrum teams adopt specific roles, create special artifacts, and hold regular ceremonies to keep things moving forward.
- **Kanban** is in a way is a simplified version of Scrum. No set roles defined in Kanban. No sprint iterations are used.

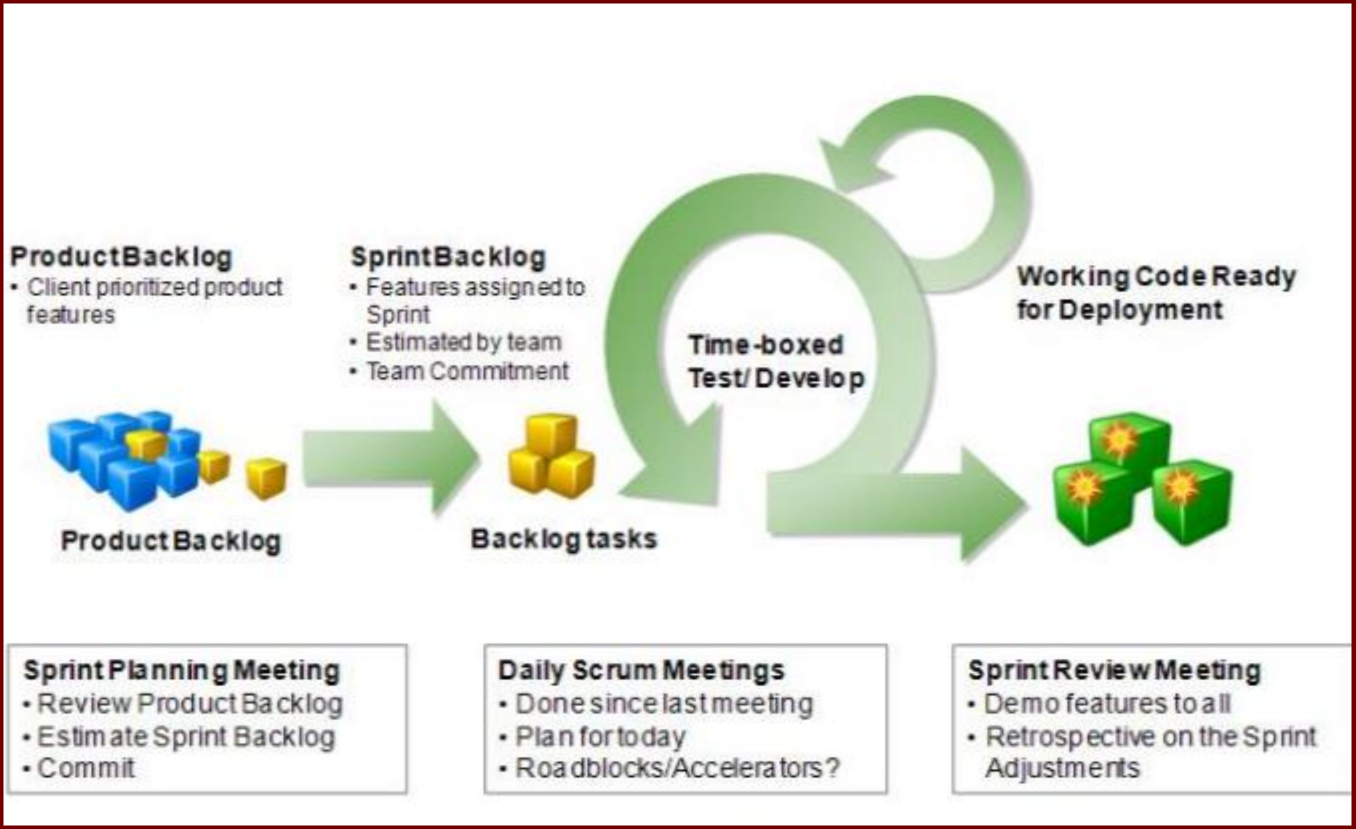
Scrum Terminology

- Sprint - short fixed-length iterations usually 2 – 4 weeks
- Product Backlog – product is described as a list of features (requirements)
- Sprint backlog – list of selected top items of Product backlog based on its priority

SCRUM Process

- The features are described in terms of user stories.
- The scrum team estimates the work associated with each story
- Team works on the defined sprint backlog
- Daily scrum meeting

SCRUM Process



ProductBacklog
- Client prioritized product features

SprintBacklog
- Features assigned to Sprint
- Estimated by team
- Team Commitment

Time-boxed Test/Develop

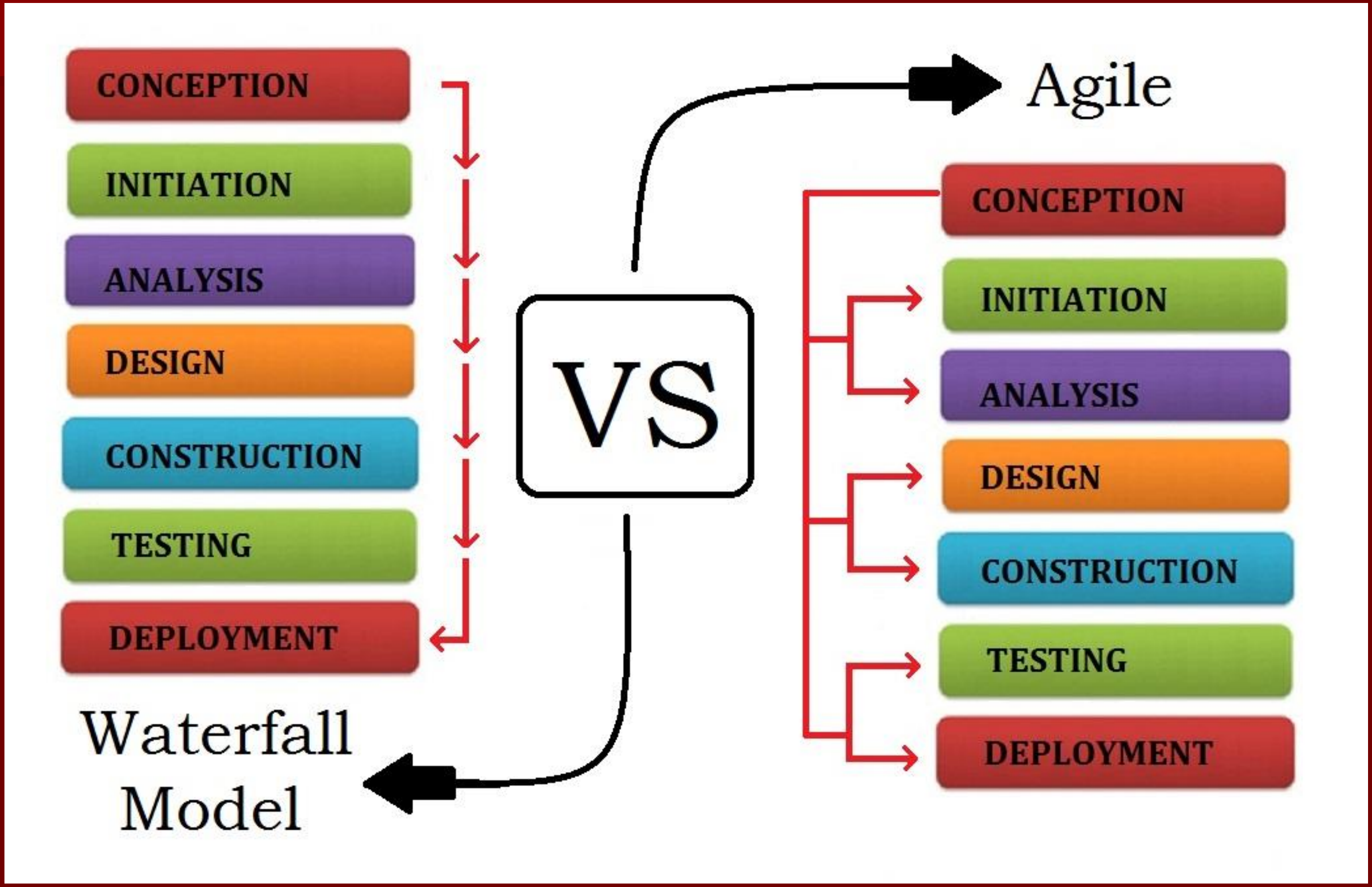
Working Code Ready for Deployment

Sprint Planning Meeting
- Review Product Backlog
- Estimate Sprint Backlog
- Commit

Daily Scrum Meetings
- Done since last meeting
- Plan for today
- Roadblocks/Accelerators?

Sprint Review Meeting
- Demo features to all
- Retrospective on the Sprint Adjustments

Waterfall vs. Agile



CONCEPTION

INITIATION

ANALYSIS

DESIGN

CONSTRUCTION

TESTING

DEPLOYMENT

Waterfall Model

VS

Agile

CONCEPTION

INITIATION

ANALYSIS

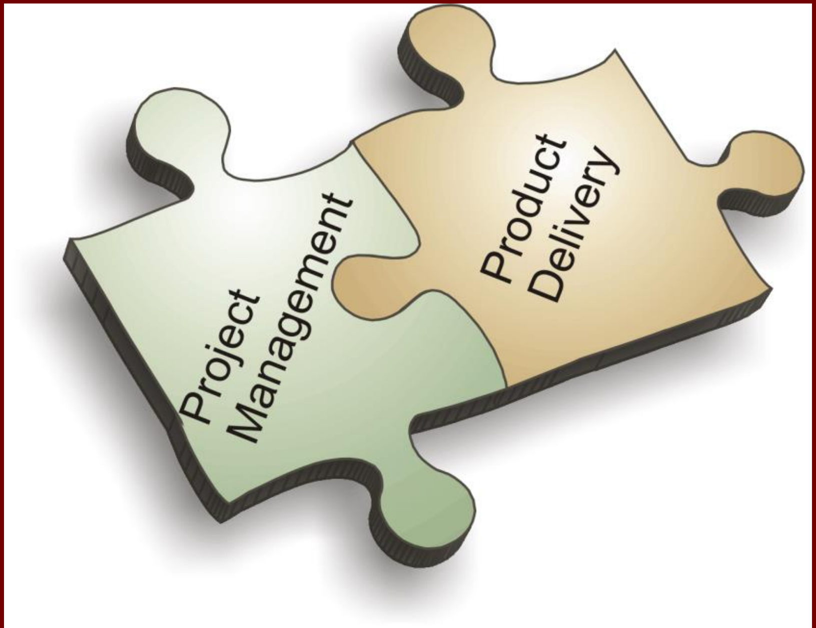
DESIGN

CONSTRUCTION

TESTING

DEPLOYMENT

Project Documentation



Project Documentation

Project Plan is a formal document to summarize business, management and financial aspects of a project.

- “Contract” between Project Manager and customers.
- It includes scope, objectives, benefits, costs, risks and plans, etc.

Project Charter/Plan

Review Project Plan and learn about your new software development project.

- What is the name of the project?
- What type of application is being developed?
- What are dependencies? Risks? Assumptions?
- What type of activities are scheduled?
- What kind of resources does the project have?
- Size of the application?
- Development effort? QA effort?

Project Plan => QA Plan Foundation

Project Plan	QA Plan
Project Statement/ Description	QA strategy, project size
Project Goals and Objectives	QA strategy, objectives
Project Scope	QA in and out of scope
Assumptions	QA assumptions, dependencies
Constraints	QA constraints: budget, technical, resources
Project Milestones	Synchronize QA milestones
Project Structure	QA resources pool
Roles and Responsibilities	Escalation path

Product Requirements

- **MRD-** Marketing Requirements Document
- **PRD-** Product Requirements Document
- **Use Cases**
  - What is it?
  - What kind of information to expect in it?
  - How will QA use it?
  - What to do if it is not available?

Product Requirements

PRD- The purpose is to clearly and unambiguously articulate the product’s purpose, features, functionality, and behavior.

MRD- The market requirements that describe the opportunity or the market need.

Use Cases

Use Cases- a format for specifying system requirements used by Business Analysts.

- Each use case represents a completed business operation performed by the user.

- From the QA perspective, we need to execute End-to-End test to make sure the requirement is implemented.

Why do I need Product Requirements?

Review Product Requirements and learn about features of your application:

- What *type of QA activities* you might need to conduct based on requirements analysis ?
- What *type of testing* is required?
- Conduct *gap analysis* to identify areas that are missing?
- Define a *list of requirements* to *clarified*.

The most important impact QA can have on product development process:

- Clarifying requirements;
- Bringing down percentage of code re-written due to the change in requirements.

No Requirements

How to test if there are no requirements?

- Use common sense;
- Compare to similar applications;
- Try exploratory testing.

Why do you test?

…to do Quality Assurance!

- Requirements Analysis
- Test Planning
- Bug Reporting
- Bug Tracking
- Test Automation
- Release Certification

Maintain and improve system!

SDLC – Environments

Tests may occur in the following hardware and software environment types:

- Development;
- QA environment;
- Staging;
- Production; etc.

Interview Questions

1. Why QA?
2. What does quality assurance mean to you?
3. Describe the QA Process.
4. When does Testing begin?
5. Why do you choose the QA career?
6. If you need to complete 20 test cases for 5 days, but after the first day you completed only 15 test cases, what will you do?
7. What is the software development life cycle?
8. How can you describe QA process in your last company?
9. How should QA engineer communicate with a developer?
10. What is the most important impact QA can have on a product development process?
11. What is use case? Write a use case for this door.
12. What is QA Methodology?
13. What are some of the differences between Agile and Waterfall? And have you used both in your previous employment?
14. What is the difference between requirements and specifications?

HOMEWORK

1. **Review class material**

2. **Prepare Interview Questions:**
   http://www.data.portnov.com/QA/Session1Questions.html

3. Quick Reference: https://portnov.net/drill/