Innominds

Agility Flexibility Relevance

Innominds SDLC for New Hire Orientation

2016-06-24

What is SDLC

- A **software development process**, also known as a **software development life-cycle** (**SDLC**), is a structure imposed on the development of a software product.
- Describe a process for planning, creating, testing, and deploying software product
- Agile and Waterfall model

Twelve Agile Principles

1. Customer satisfaction by rapid delivery of useful software
2. Welcome changing requirements, even late in development
3. Working software is delivered frequently (weeks rather than months)
4. Close, daily cooperation between business people and developers
5. Projects are built around motivated individuals, who should be trusted
6. Face-to-face conversation or conversation using collaboration tool (Such as WebEX, GoToMeeting etc.) is the best form of communication to ensure that all members are present at the same time.

Twelve Agile Principles

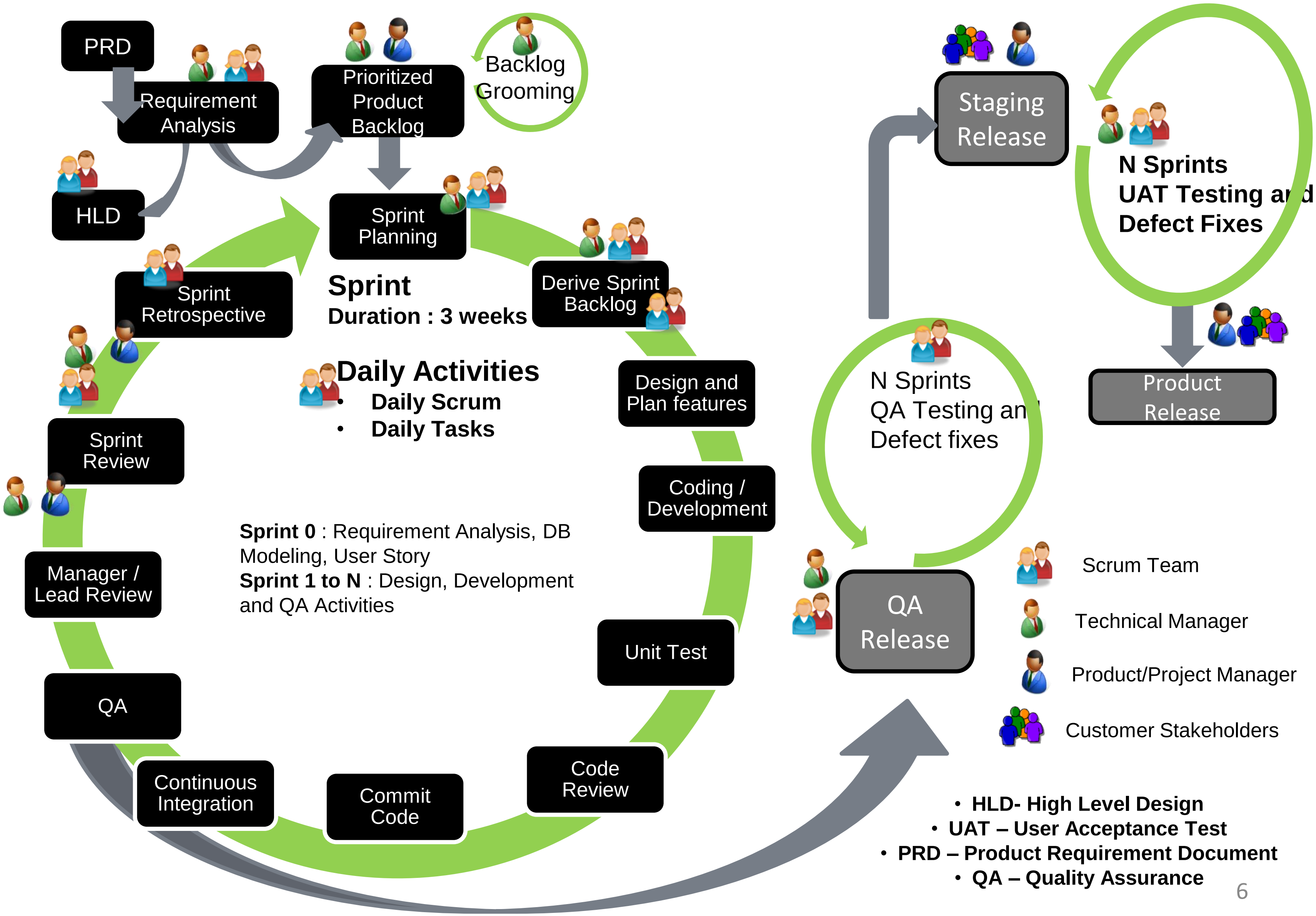
7. Working software is the principal measure of progress
8. Sustainable development, able to maintain a constant pace
9. Continuous attention to technical excellence and good design
10. Simplicity—the art of maximizing the amount of work not done—is essential
11. Self-organizing teams
12. Regular adaptation to changing circumstances

Scrum Agile Method

- **Scrum** is an iterative and incremental agile software development framework.
- Scrum adopts an empirical approach—accepting that the problem cannot be fully understood or defined, focusing instead on maximizing the team's ability to deliver quickly and respond to emerging requirements.

Innominds

Innominds -Agile Engineering Process



Sprint
Duration : 3 weeks

Daily Activities
- **Daily Scrum**
- **Daily Tasks**

Sprint 0 : Requirement Analysis, DB Modeling, User Story
Sprint 1 to N : Design, Development and QA Activities

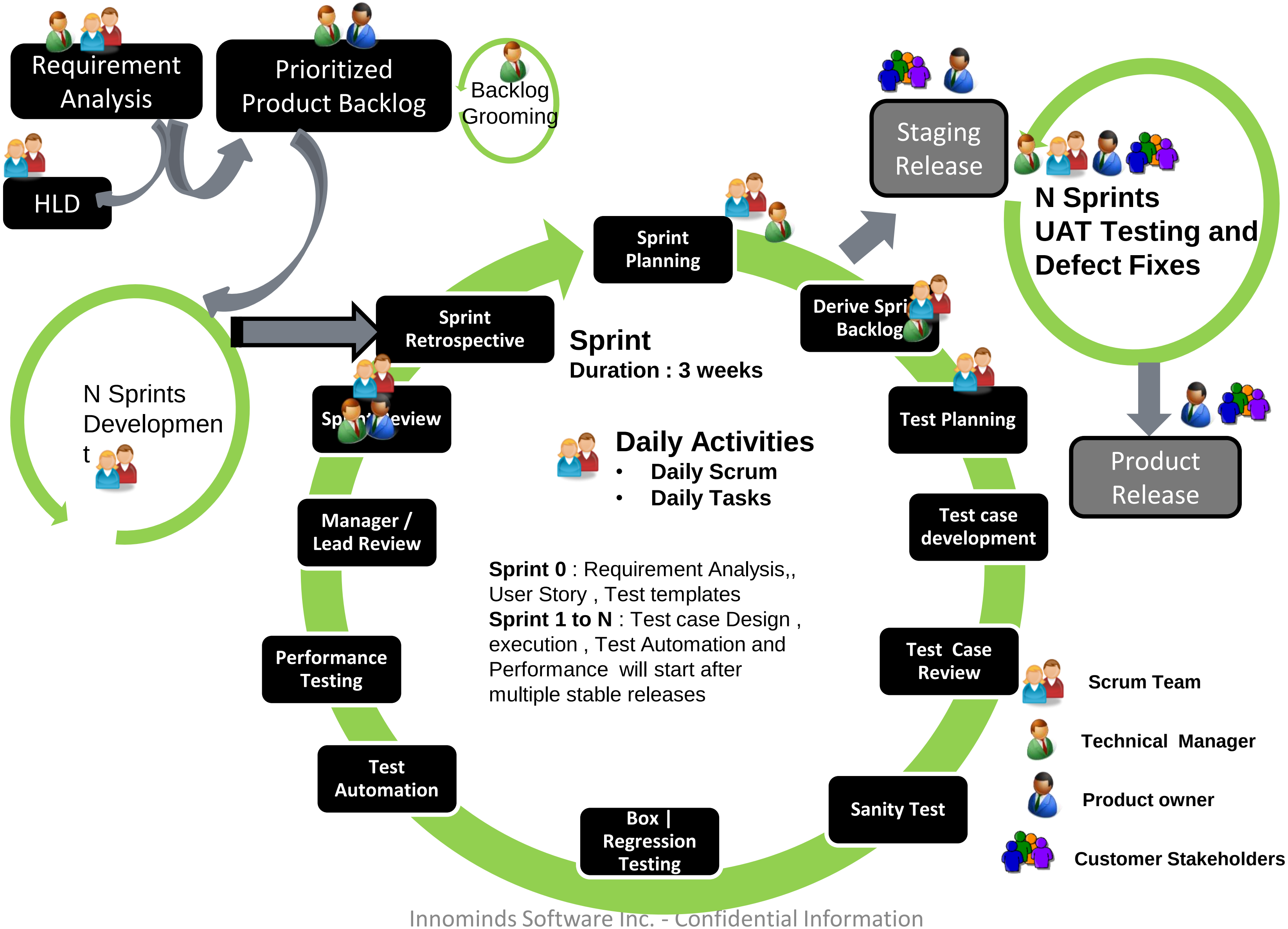
N Sprints
QA Testing an
Defect fixes

N Sprints
UAT Testing ar d
Defect Fixes

Scrum Team
Technical Manager
Product/Project Manager
Customer Stakeholders

- **HLD- High Level Design**
- **UAT – User Acceptance Test**
- **PRD – Product Requirement Document**
- **QA – Quality Assurance**

Innominds -Agile QA Process (2/2)



Sprint

Duration : 3 weeks

Daily Activities

- **Daily Scrum**
- **Daily Tasks**

Sprint 0 : Requirement Analysis,, User Story , Test templates
Sprint 1 to N : Test case Design , execution , Test Automation and Performance will start after multiple stable releases

Scrum Team

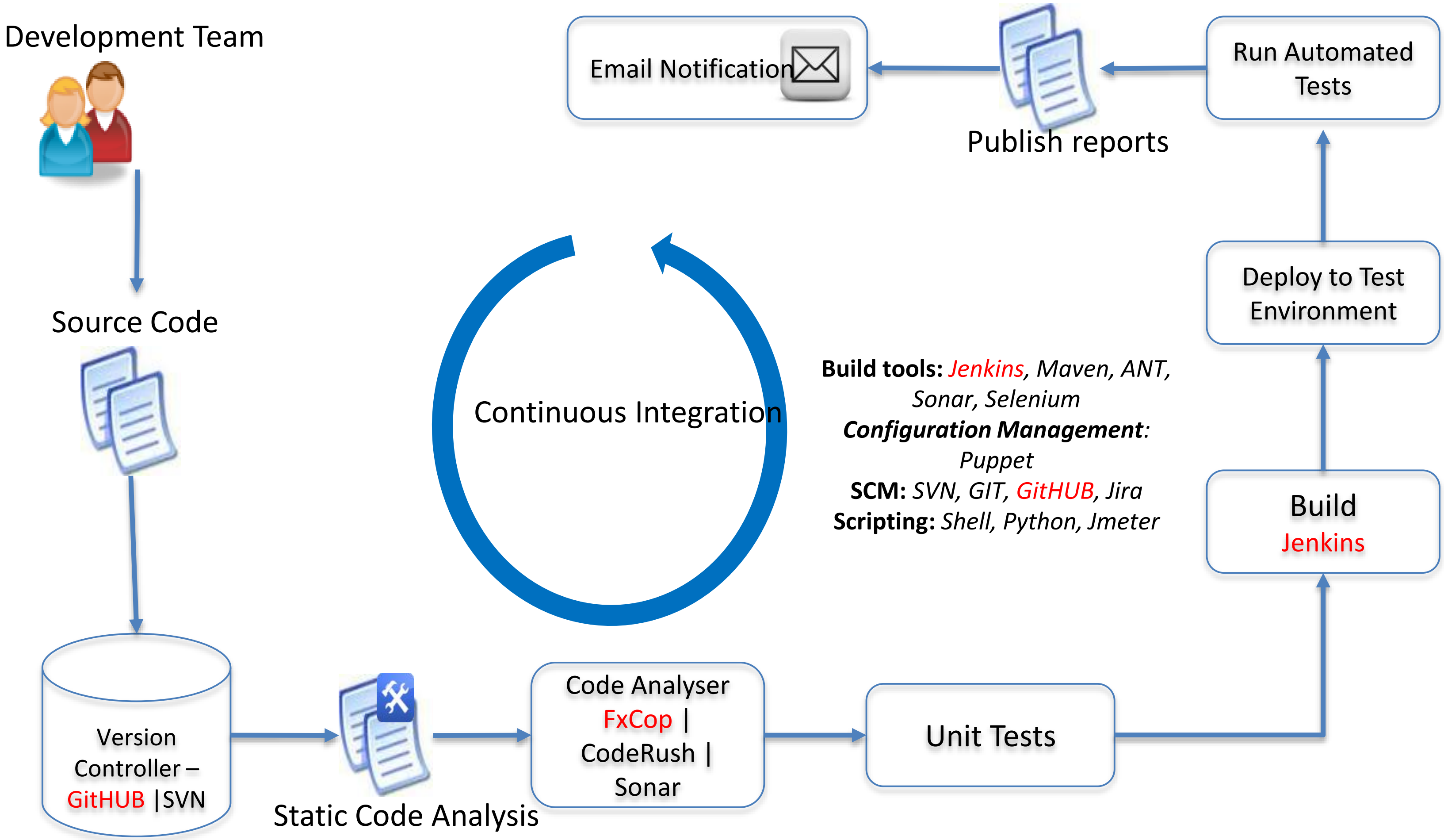
Technical Manager

Product owner

Customer Stakeholders

SDLC - Continuous Integration

Innominds

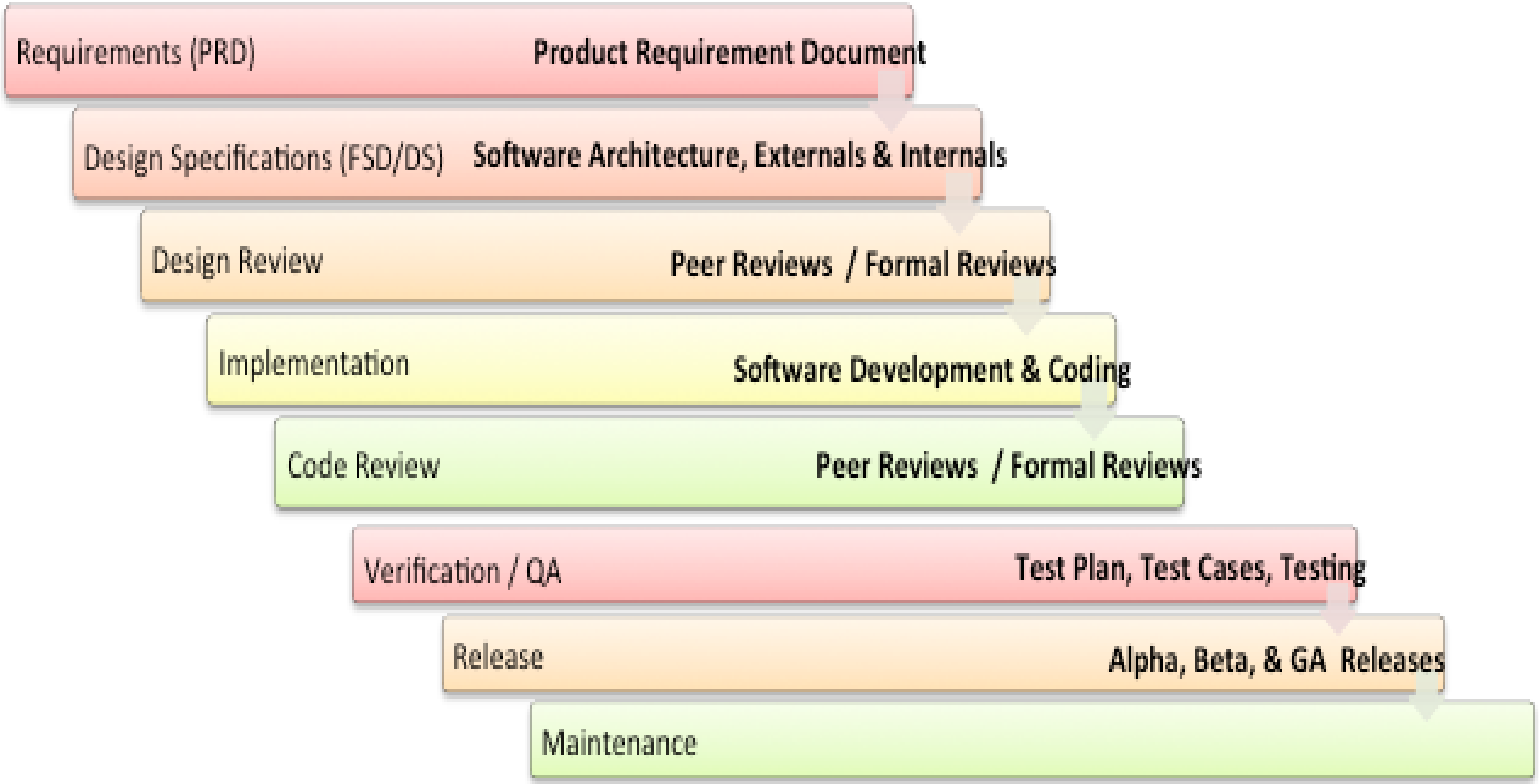


Build tools: *Jenkins, Maven, ANT, Sonar, Selenium*
Configuration Management: *Puppet*
SCM: *SVN, GIT, GitHUB, Jira*
Scripting: *Shell, Python, Jmeter*

SDLC Metrics

	Project Level	Jan-16	Feb-16	Mar-16
# Lines of code written				
# Code reviews				
# Unit test cases				
# New classes (if any)				
# New Features added (if any)				
# Total number of Test Cases (Cumulative)				
# Total Test cases automated (Cumulative)				
# Bugs opened				
# Bugs closed (QA Metric)				
# Bugs Resolved/Fixed (Dev Metric)				
# Bugs Reopened				
# Bugs did not require Fix (QA Metric)				
# Manual Test cases Executed (All Runs)				
# Test cases Passed (Latest Run)				
# Test cases Failed (Latest Run)				
# Dev effort in Person days				
# QA effort in Person days				

Waterfall model



Requirements (PRD) Product Requirement Document

Design Specifications (FSD/DS) Software Architecture, Externals & Internals

Design Review Peer Reviews / Formal Reviews

Implementation Software Development & Coding

Code Review Peer Reviews / Formal Reviews

Verification / QA Test Plan, Test Cases, Testing

Release Alpha, Beta, & GA Releases

Maintenance

* Though “pure” Waterfall discourages revisiting and revising any prior phase in practice, people do modify previous phases time to time knowing the cost implications of doing such changes.

Things to note

1. **First Believe in methodology** and best practices to follow and reap its benefits.
2. **Design before Coding**: worthwhile to avoid rework and ensures the design covers all aspects
3. **Coding**: Follow the [standard coding guidelines]()
4. **Code reviews a MUST** to ensure quality of code written
   - Allocate time for code reviews before code check-in
   - Can be done with your peer or team lead
   - Can be done before or after the Unit testing.
5. **Start Work on HLD and detailed design ASAP**: start work (HLD and detailed design) on the sections that are complete and certain to remain unchanged.

Things to note

6. **Parallelize tasks as possible**: Try to do things in parallel as much as possible
7. **Integration testing**: Do integration testing after all unit tested modules to ensure nothing is broken after integration
8. **Development Quality Metrics:** keep a track of a) #of lines of code written, b) #of Unit tests in a day
9. **Testing performance metrics:** Fill up the Testing Progress Report and efficiency reports every day or week as possible
10. **Tools/Frameworks:** Use the necessary tools to enforce processes like code review. As of now, Sonar with PMD plugin is decided for Java and fabricator is decided for PHP.
11. Explore Innominds Architecture

Q&A