Software Development Life Cycle

Agenda:

1. Overview
2. History
3. Waterfall. Incremental model
4. RUP. Iterative model
5. Kanban
6. Agile
7. Q&A

Overview

A software development lifecycle is essentially a series of steps, or phases, that provide a model for the development and lifecycle management of an application or piece of software.

Wiki:

In software engineering, a software development methodology (also known as a system development methodology, software development life cycle, software development process, software process) is a division of software development work into distinct phases (or stages) containing activities with the intent of better planning and management. It is often considered a subset of the systems development life cycle.

History

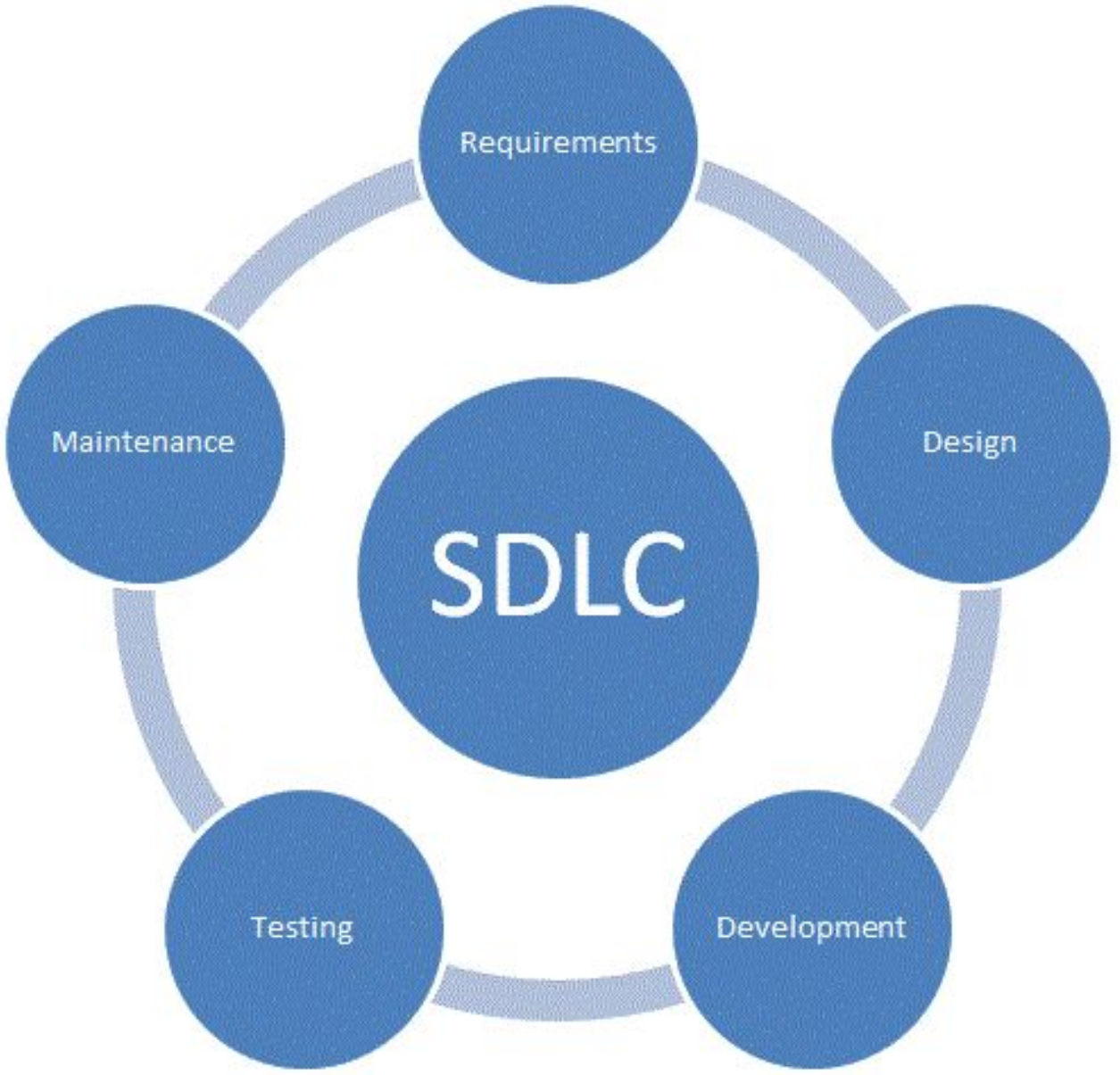
50’s – 60’s first known process

70’s – Waterfall

80’s – Kanban

1996 – RUP

2001 - Agile



Software development process

Why should we care about process?

Historical problems:

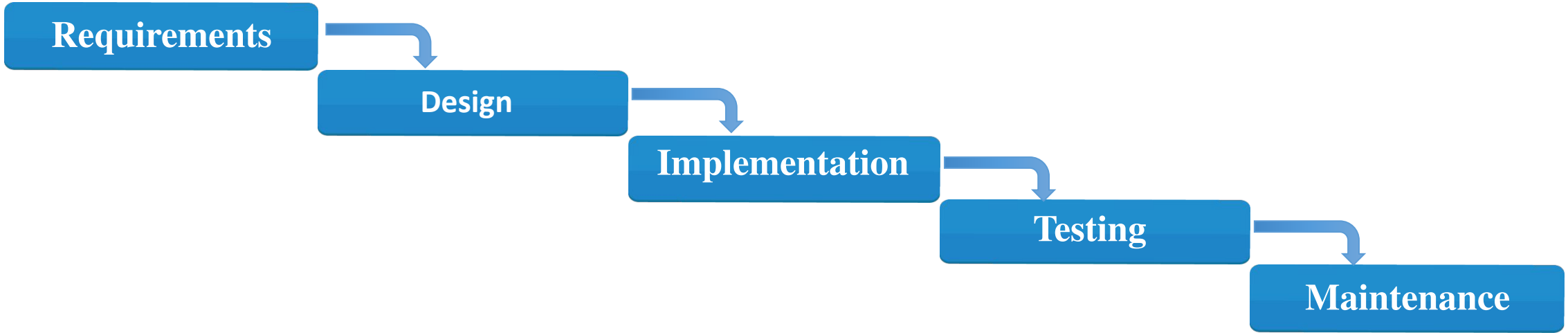
- Time evaluation

- Budget evaluation

- Documentation and support

Waterfall

Essentially, Waterfall is a framework for software development in which development proceeds sequentially through a series of phases, starting with system requirements analysis and leading up to product release and maintenance.

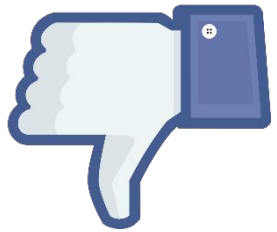


Waterfall

Simple and easy to understand and use

Process and results are well documented

Phases are processed and completed one at a time



Increased development time

System must be defined up front

No working software is produced until late during the life cycle.

High amounts of risk and uncertainty.

Difficult to measure progress within stages

Incremental model

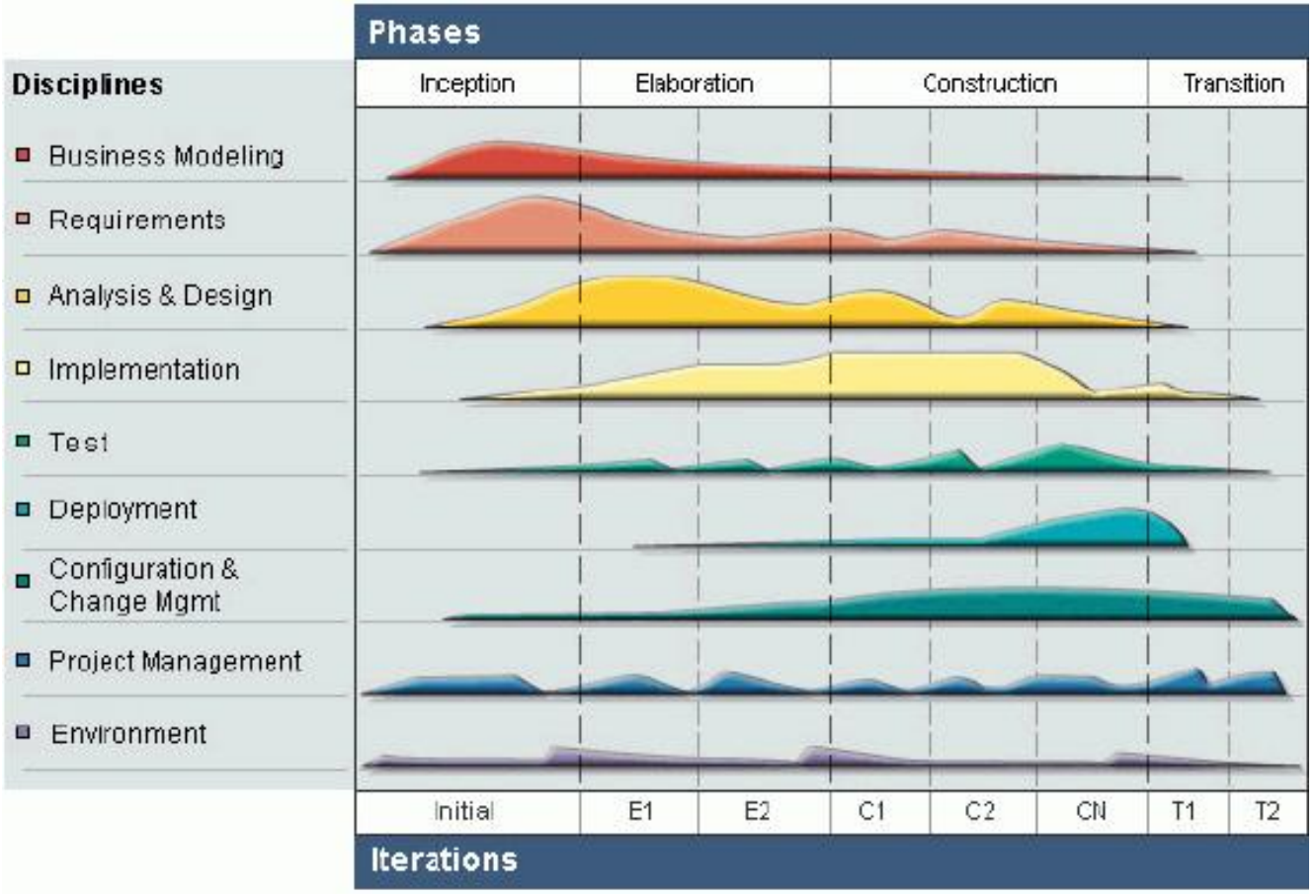
In incremental model the whole requirement is divided into various builds. Multiple development cycles take place here, making the life cycle a “multi-waterfall” cycle. Cycles are divided up into smaller, more easily managed modules. Each module passes through the requirements, design, implementation and testing phases.

Iterative model

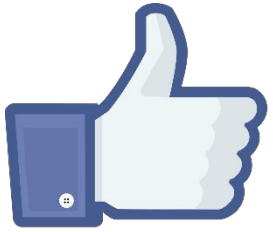
An iterative life cycle model does not attempt to start with a full specification of requirements. Instead, development begins by specifying and implementing just part of the software, which can then be reviewed in order to identify further requirements. This process is then repeated, producing a new version of the software for each cycle of the model.

RUP

Rational Unified Process (RUP) is an object-oriented and Web-enabled program development methodology. RUP establishes four phases of development, each of which is organized into a number of separate iterations that must satisfy defined criteria before the next phase is undertaken.

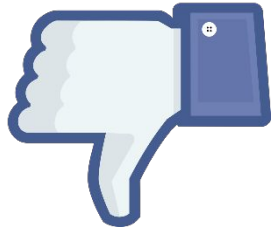


RUP



We can get the reliable user feedback

Less time is spent on documenting and more time is given for designing



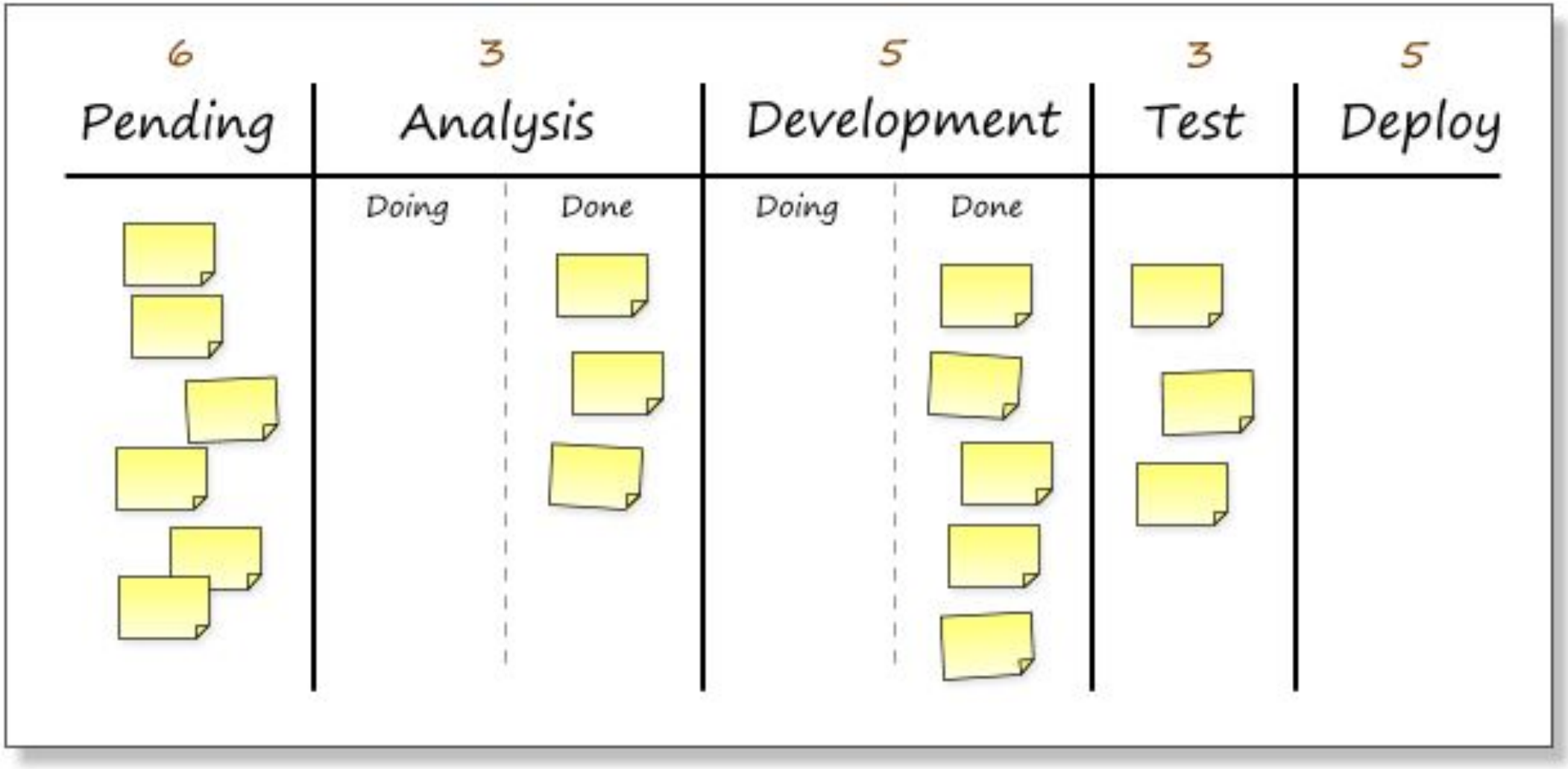
Costly system architecture or design issues may arise because not all requirements are gathered up front for the entire lifecycle

On cutting edge projects which utilise new technology, the reuse of components will not be possible

Kanban

Kanban is a new technique for managing a software development process in a highly efficient way. Kanban underpins Toyota's "just-in-time" (JIT) production system. A software development process can be thought of as a pipeline with feature requests entering one end and improved software emerging from the other end.

Kanban

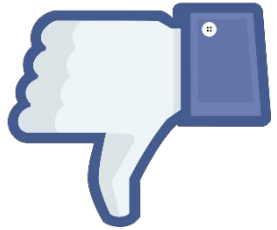

6
Pending
3
Analysis
Doing
Done
5
Development
Doing
Done
3
Test
5
Deploy

Kanban

Visualization Workflow

Limit Work-in-Progress

Provides flexibility in production



Abused limits

Possible delays between phases

Less effective in shared-resource situations

Agile

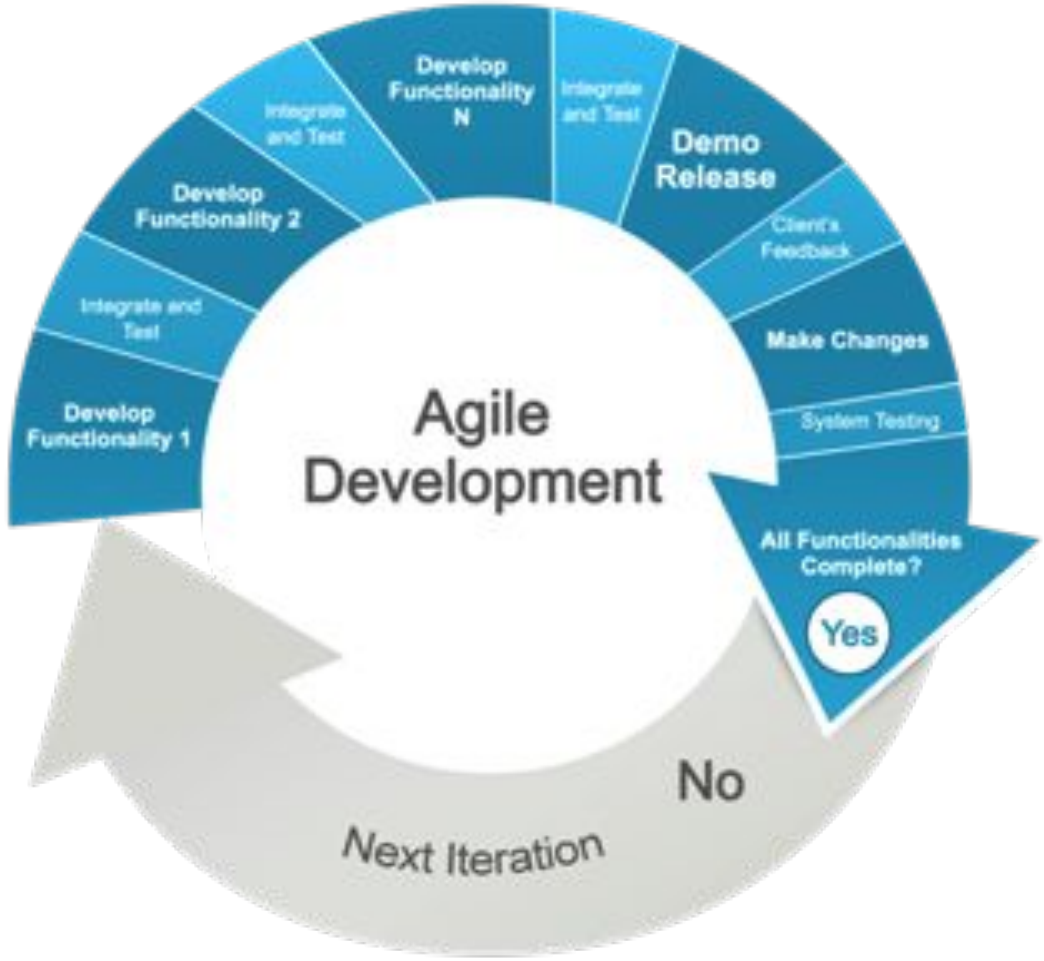
We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

Individuals and interactions over
processes and tools

Working software over
comprehensive documentation

Customer collaboration over
contract negotiation

Responding to change over
following a plan



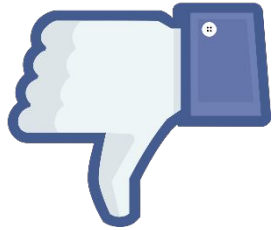
Agile

Changes

Team interaction

Continuous improvement

Encourages active involvement and interaction from key project stakeholders



Planning (target delivery date consisting of x)

Team members must be highly skilled / cross skilled

Documentation


ASK ME