Announcements

- Group Mentors Assigned- You should have received ab email
- Extra Credit Opportunity in today’s class
- Quiz 2 will release on 9/15 from 7 AM – 11:59 PM.
    - 10 minutes; Can take once; Using Honorlock;
    - Open physical notes; or browser tab restricted to Canvas, class website, lecture notes
    - Pool of questions; 15 random questions per student;
    - Based on class notes - GIT, life cycle model, RE, AJAX, REST, Maven

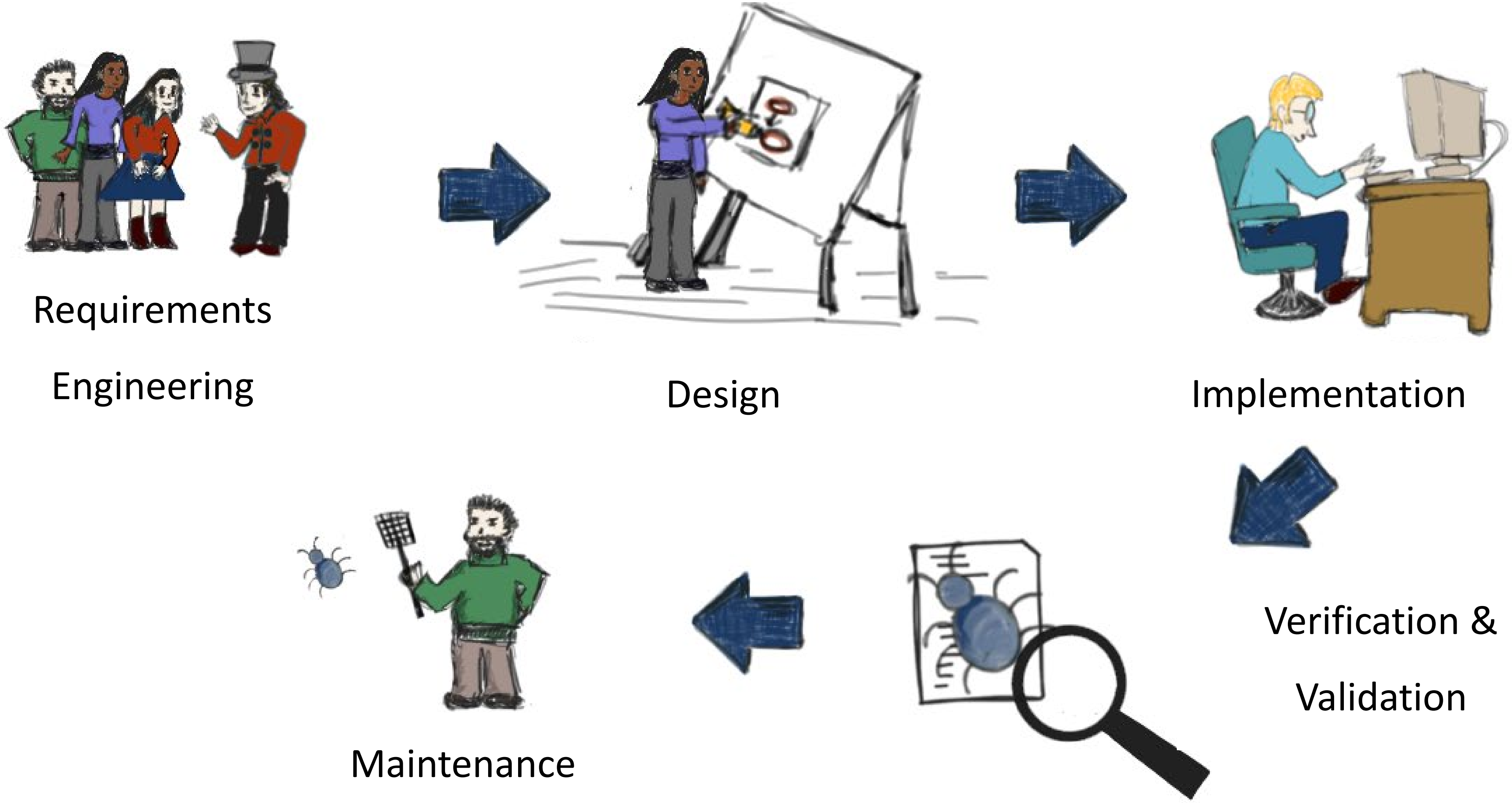
CS3300 Introduction to Software Engineering

Lecture 03: Life Cycle Models

Nimisha Roy ▸ nroy9@gatech.edu

Slides adapted from Alessandro Orso

Traditional Software Phases



Requirements Engineering

Design

Implementation

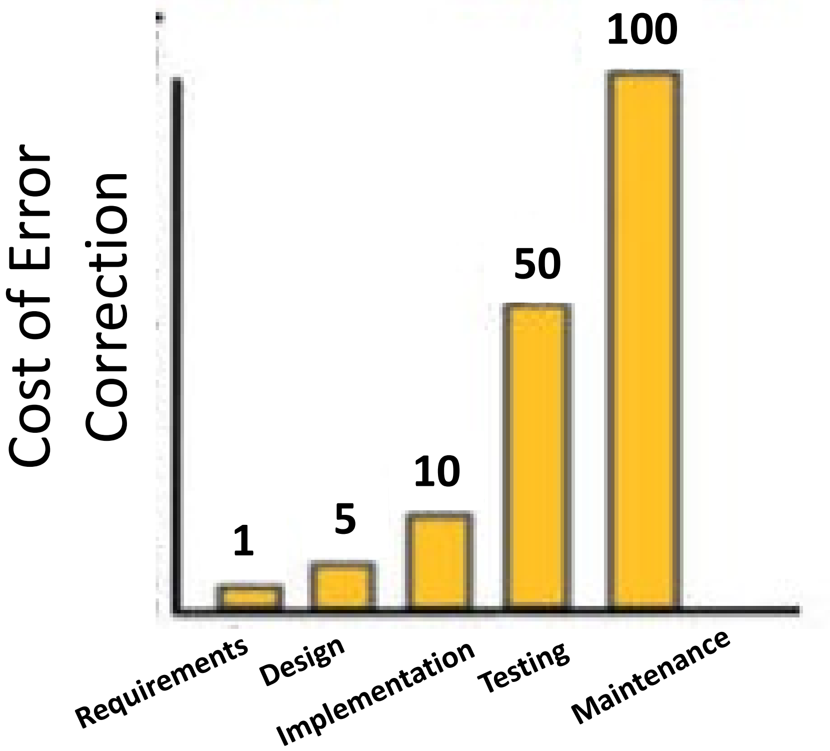
Verification & Validation

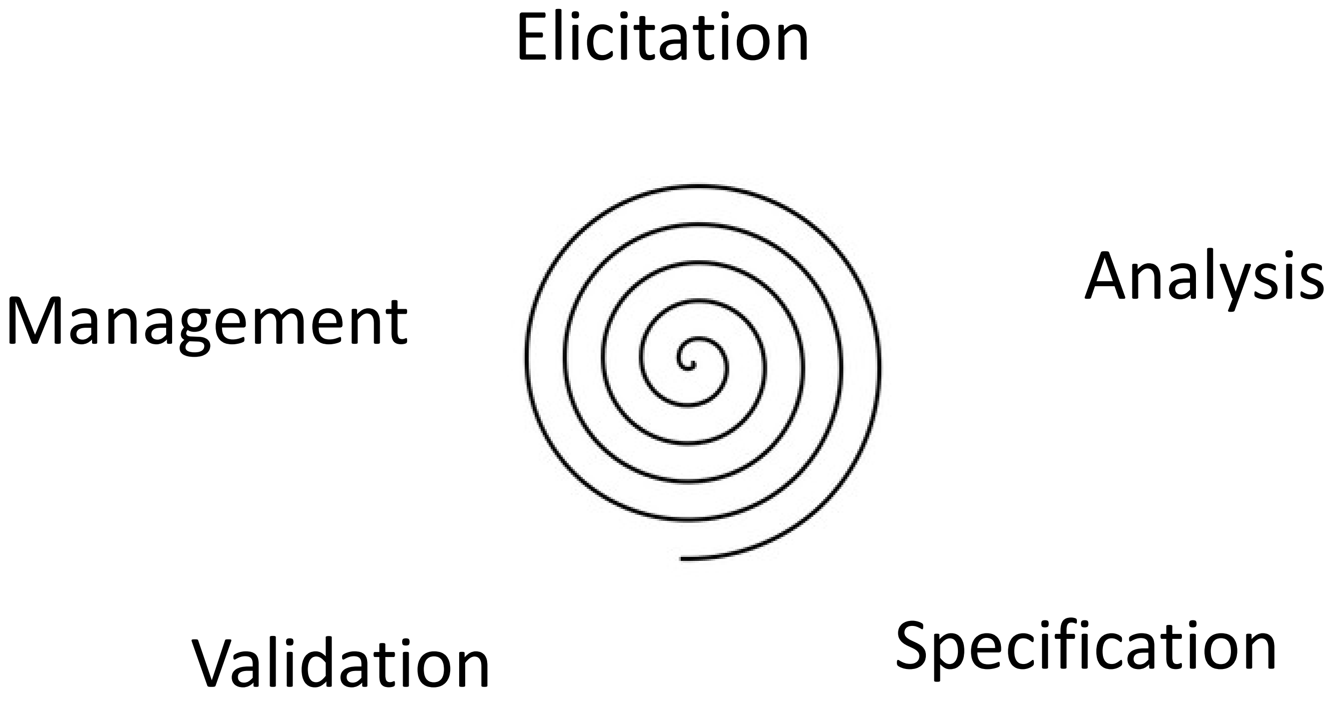
Maintenance

Requirements Engineering

RE is the process of establishing the needs of stakeholders that are to be solved by software

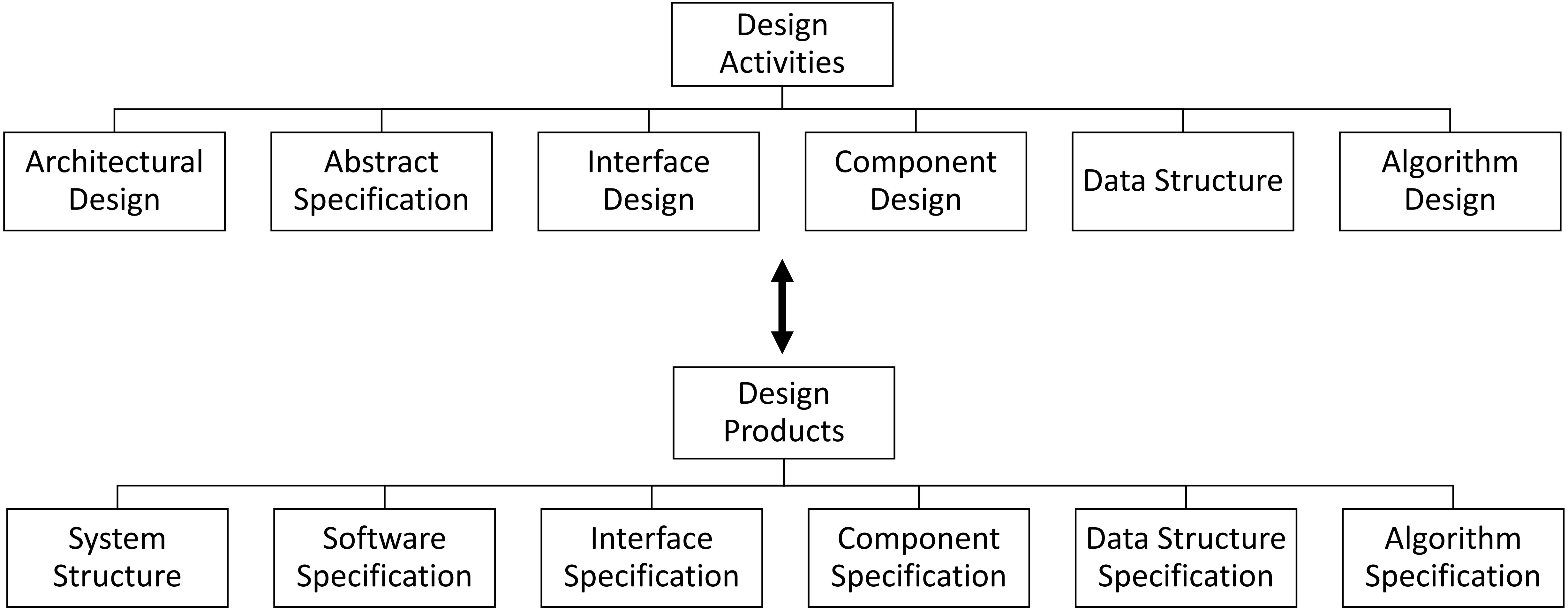
Cost of Late Correction





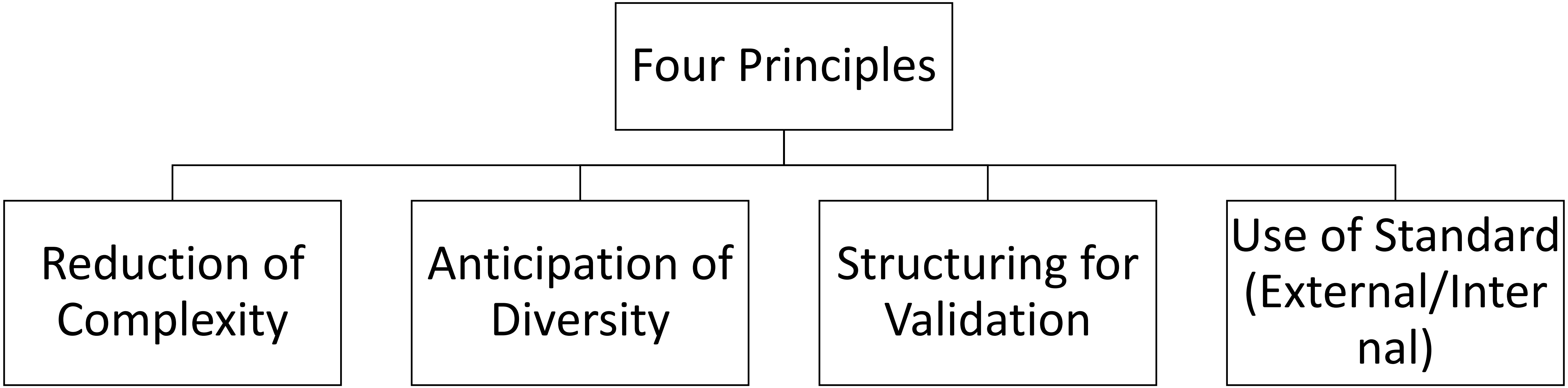
Design

Phase where requirements are analyzed in order to produce a description of the internal structure and organization of the system. Basis for construction of the actual system



Implementation

Phase where we take care of realizing the design of the system and create a natural softer system



Four Principles

Reduction of Complexity

Anticipation of Diversity

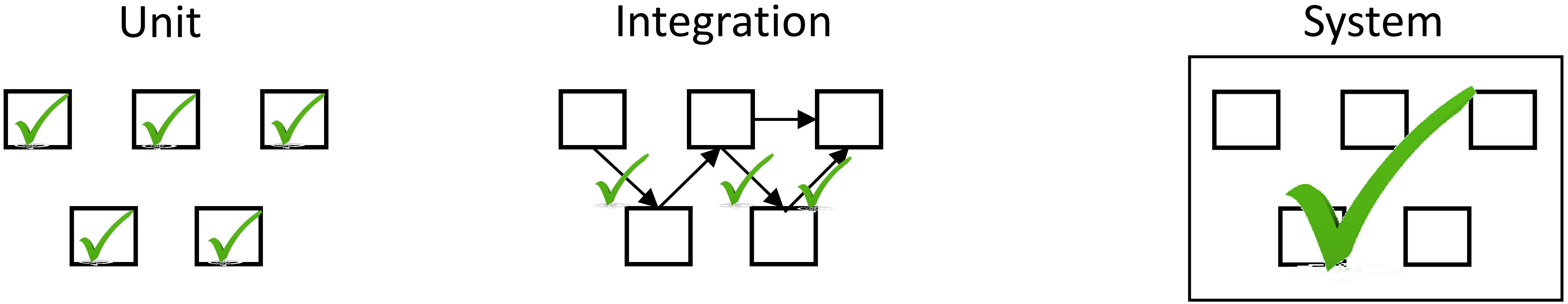
Structuring for Validation

Use of Standard (External/Internal)

Verification & Validation

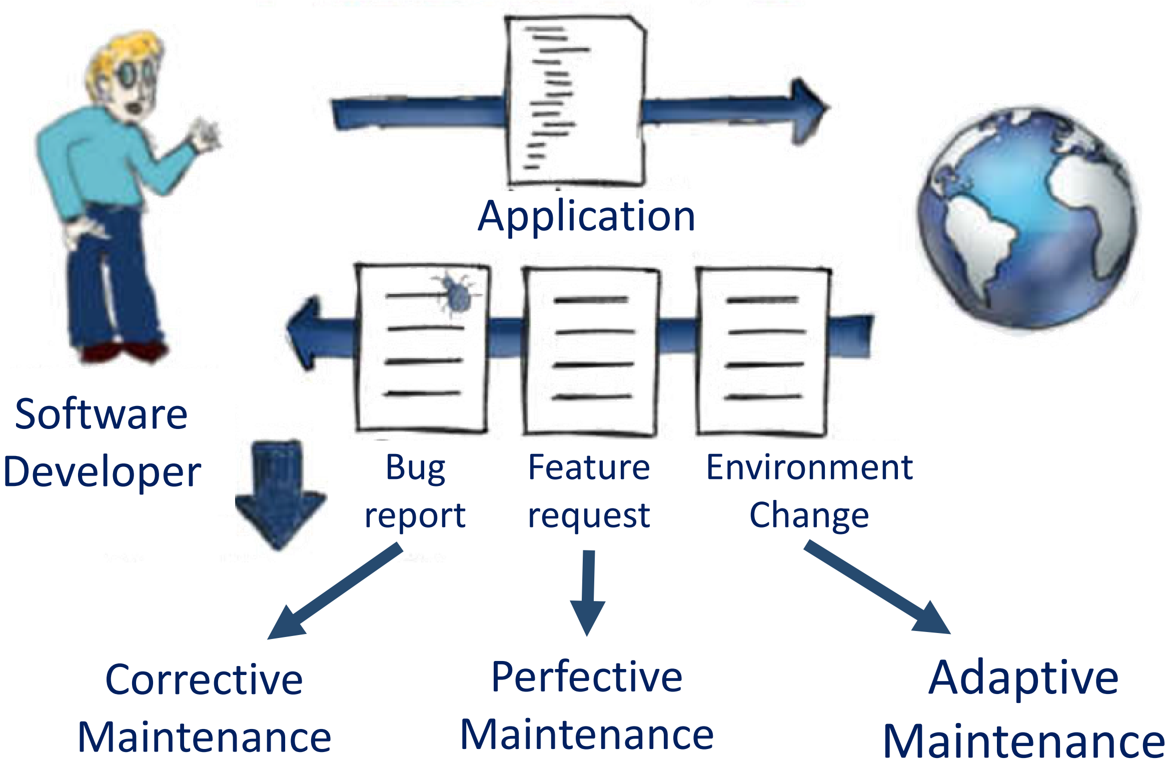
Phase that aims to check that software system meets its specifications and fulfils its intended purpose

Verification: did we build the system right?
Validation: did we build the right system?



Maintenance

Once Software released to final users and in operation, many things can happen: environment change -new libraries, new systems, additional functionality requests, bug reports



Application

Software Developer

Bug report

Feature request

Environment Change

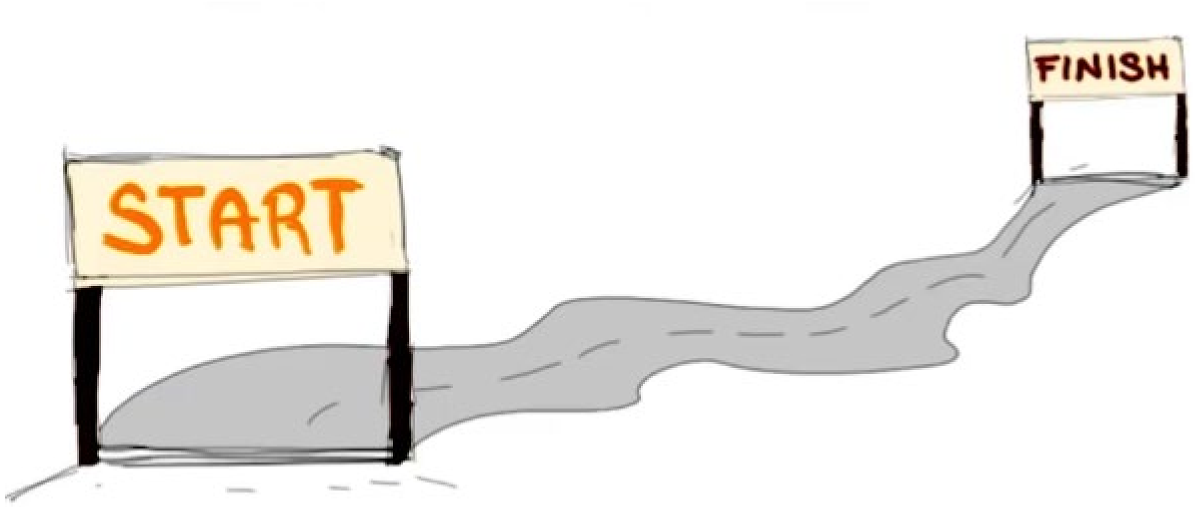
Corrective Maintenance

Perfective Maintenance

Adaptive Maintenance

- Maintenance is a fundamental and expensive phase
- Regression testing – retesting a modified version of software before release, no introduction of new errors

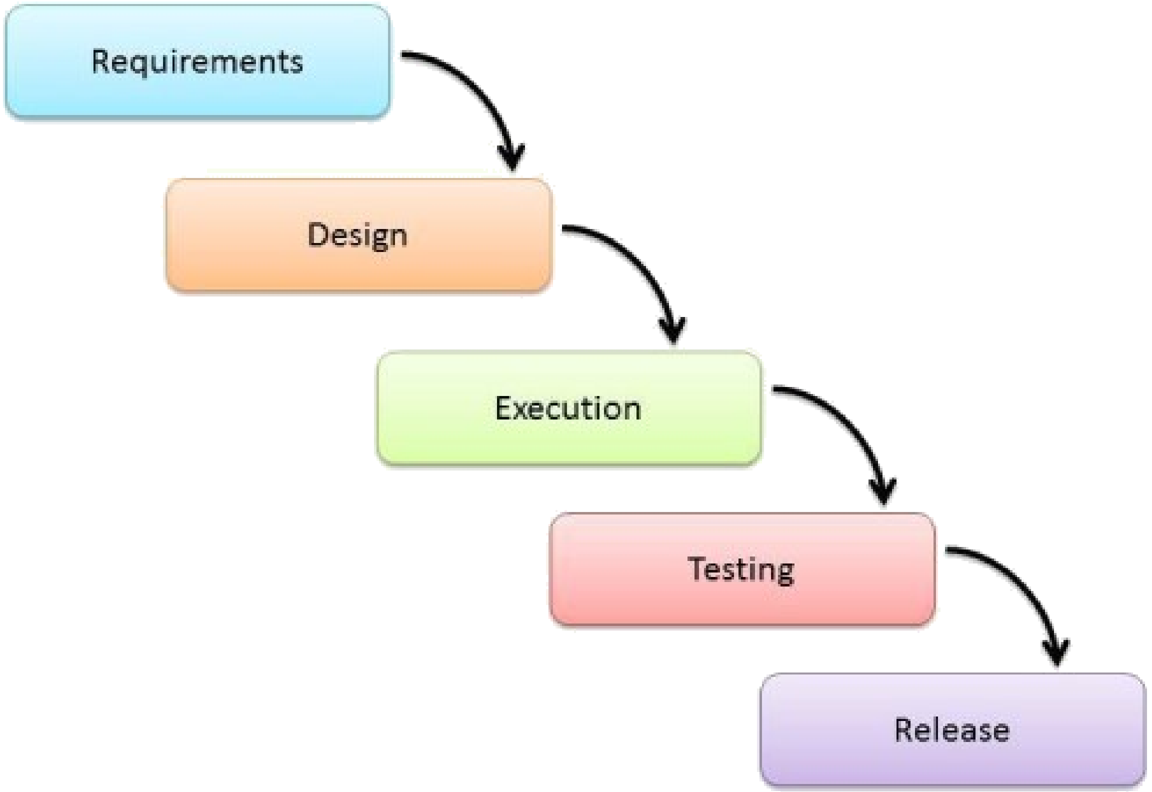
Software Process Model/ Life Cycle Model



Functions:

- Order of activities
- Transition Criteria between Activities
- What should we do next and for how long?

Waterfall Method


Requirements

Design

Execution

Testing

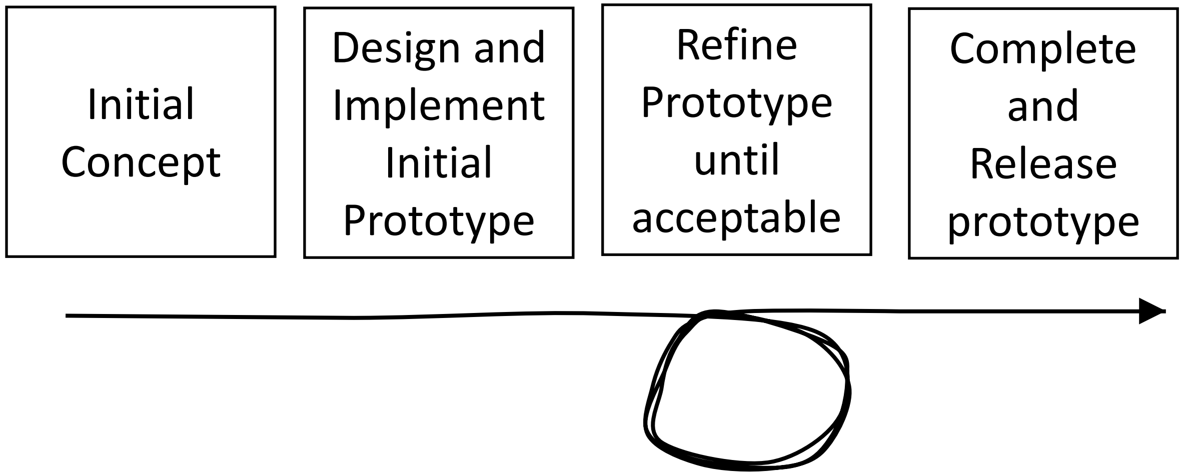
Release

👍 Early Error Detection

👎 No Flexibility

- Project progresses in an orderly sequence of steps
- Pure Waterfall model performs well for software products with a stable product definition- well known domain, technologies involved, Request for Proposals (RFP)
- Waterfall method finds errors in early local stages
- Not flexible- not for projects where requirements change, developers not domain experts, or technology used are new and evolving

Evolutionary Prototyping



Initial Concept

Design and Implement Initial Prototype

Refine Prototype until acceptable

Complete and Release prototype

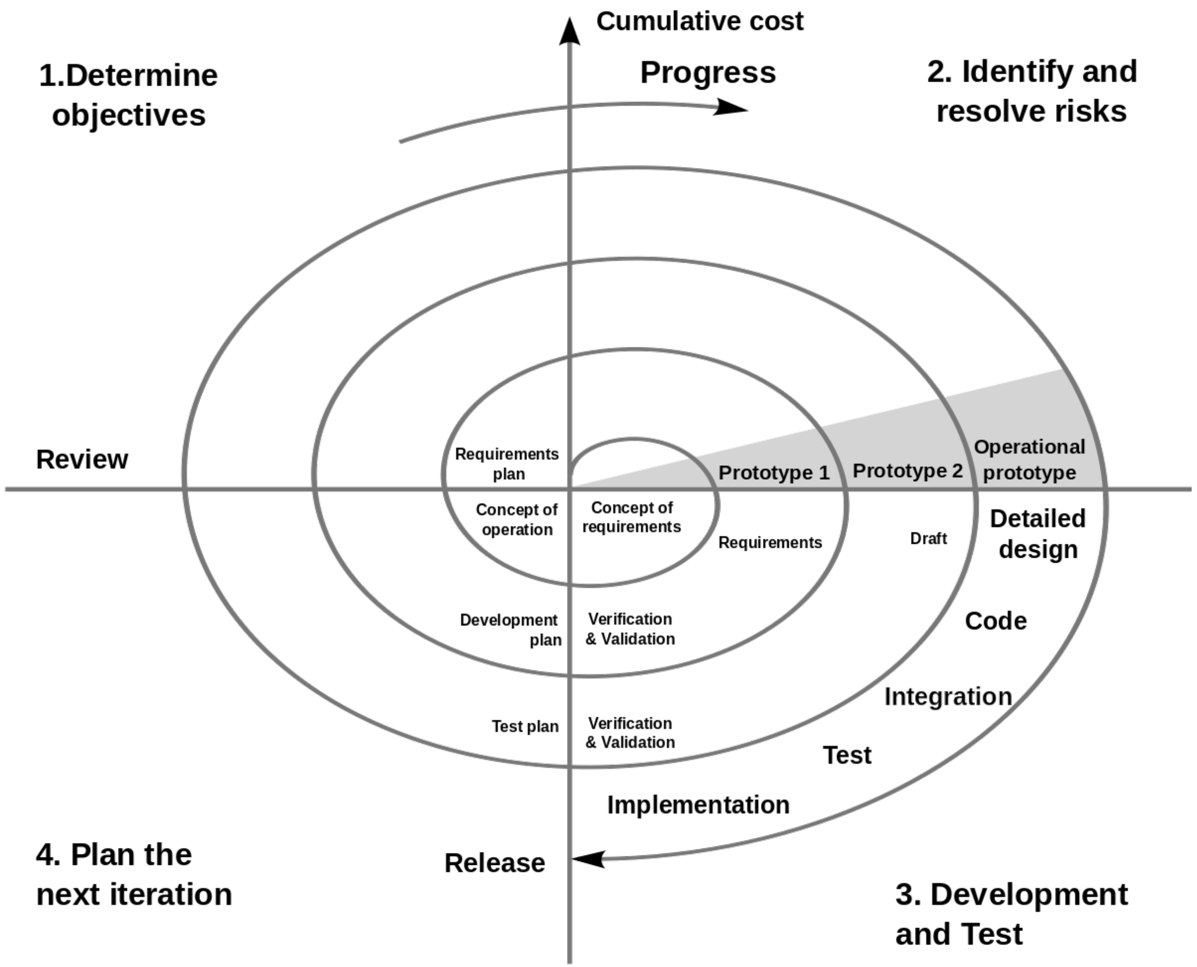
👍 Immediate feedback
Helps Requirements understanding

👎 Difficult to Plan
Can deteriorate to code-and-fix

- Prototypes that evolve into the final system through an iterative incorporation of user feedback.
- Ideal when not all requirements are well-understood. System keeps evolving based on customer feedback
- Developers start by developing the parts of the system that they understand, instead of developing a whole system. Partial system is then shown to the customer and the customer feedback is used to drive the next iteration, in which either changes are made to the current features or new features are added.

Spiral Method

Incremental risk-oriented lifecycle model with 4 main phases



👍

Risk Reduction

Functionality can be added

Software produced early, Early feedback

👎

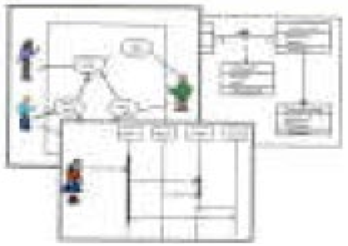
Specific Expertise

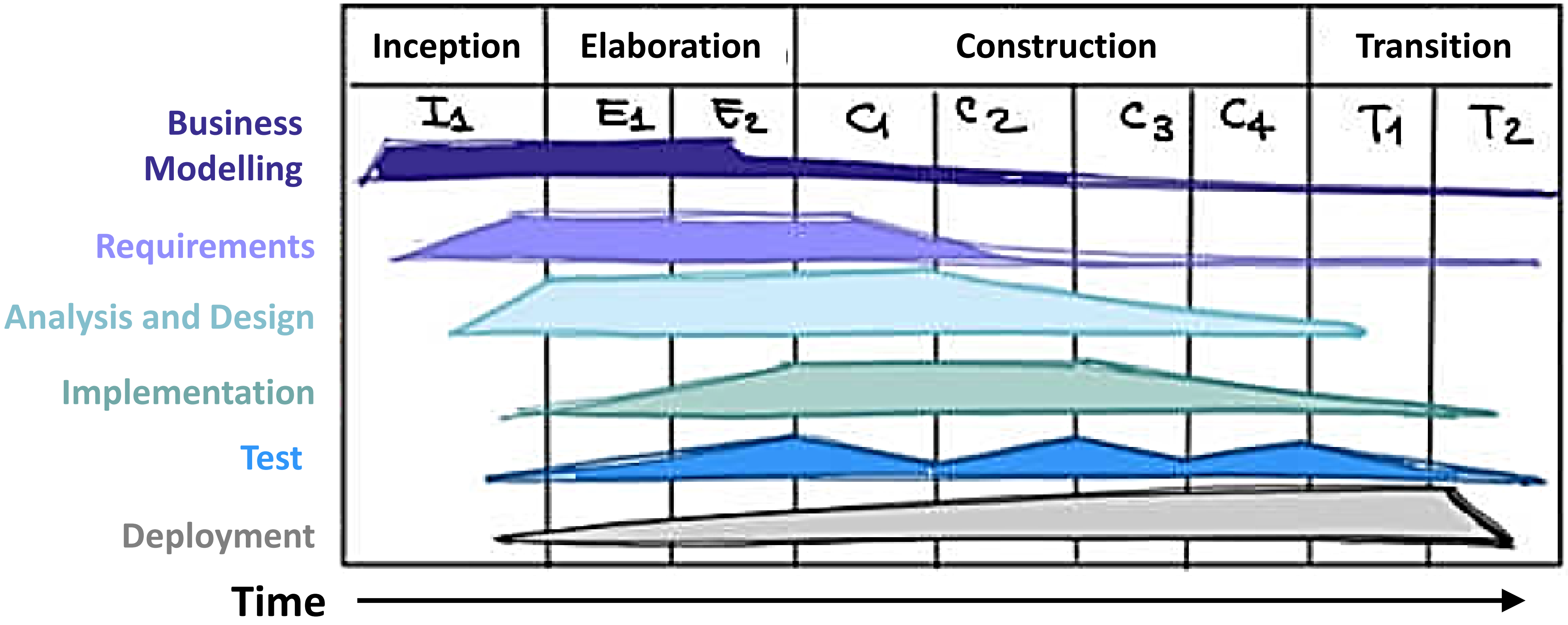
Highly dependent on risk analysis

Complex, Costly

https://melsatar.blog/2012/03/15/software-development-life-cycle-models-and-methodologies/

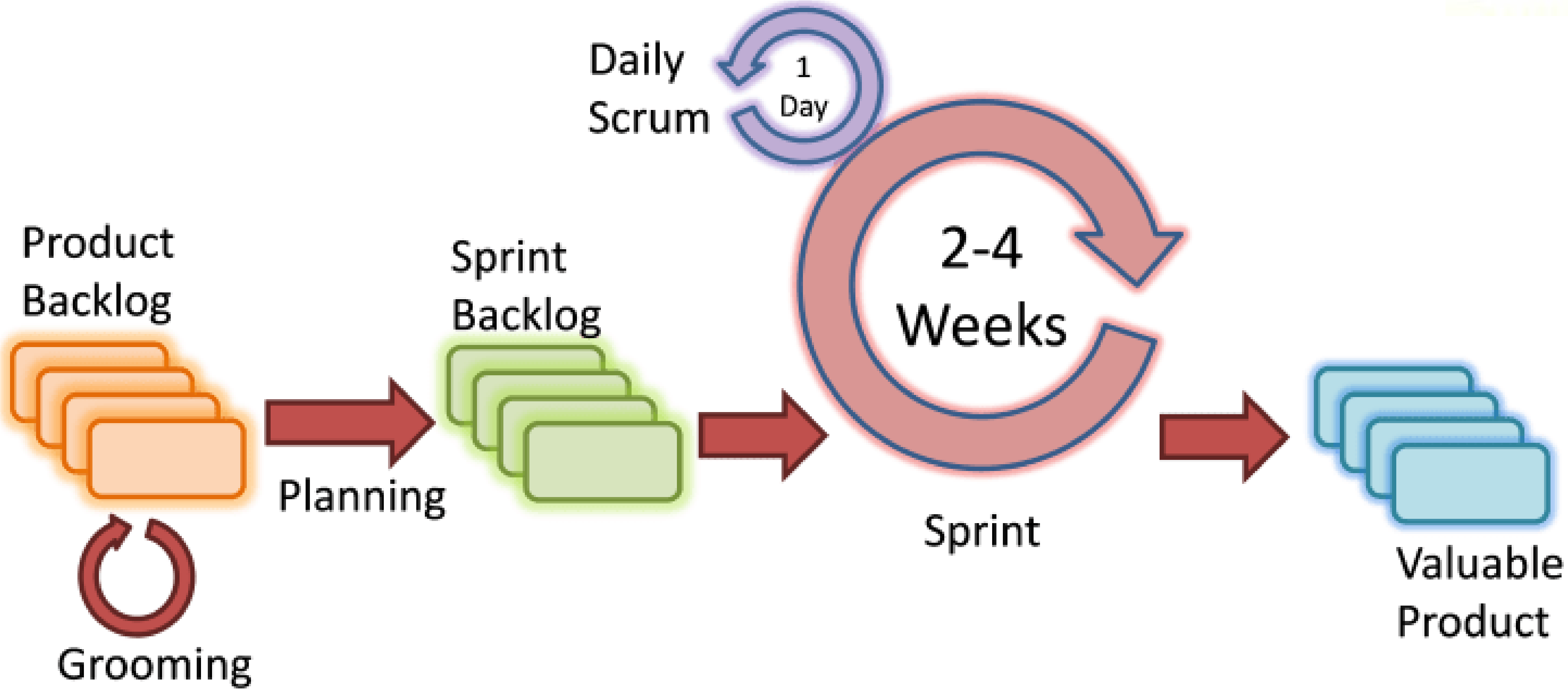
Rational Unified Process (RUP)





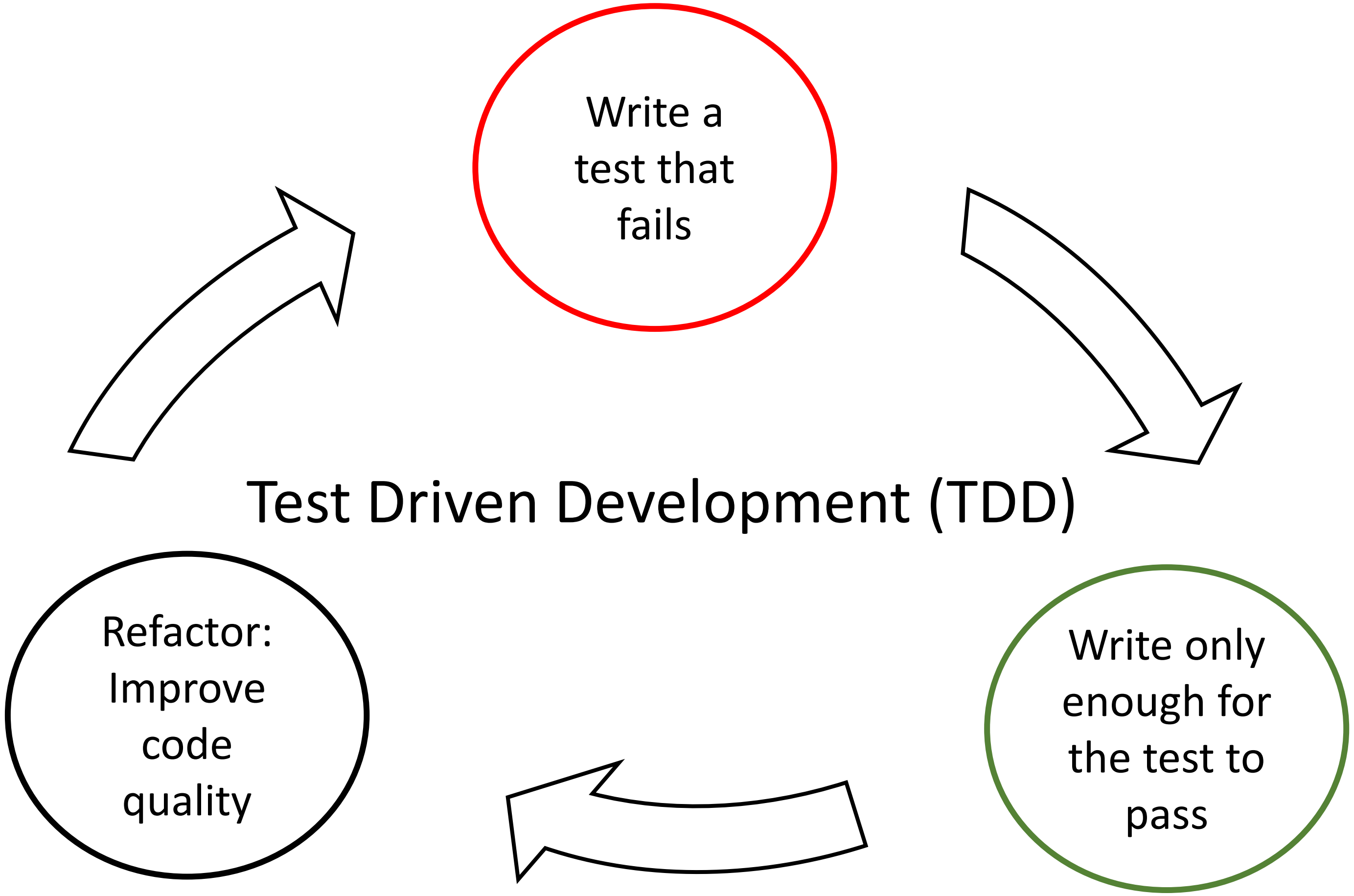
- Popular Process based on UML. Works iteratively, performs 4 phases in each iteration
- Inception phase: Scope the system - Scope of project, domain, initial cost, budget estimates
- Elaboration phase: domain analysis and basic architecture
- Construction phase: Bulk of development
- Transition: From development to production, available to users

Agile - Scrum



Highly iterative and incremental development process

Agile - XP



Write a test that fails

Write only enough for the test to pass

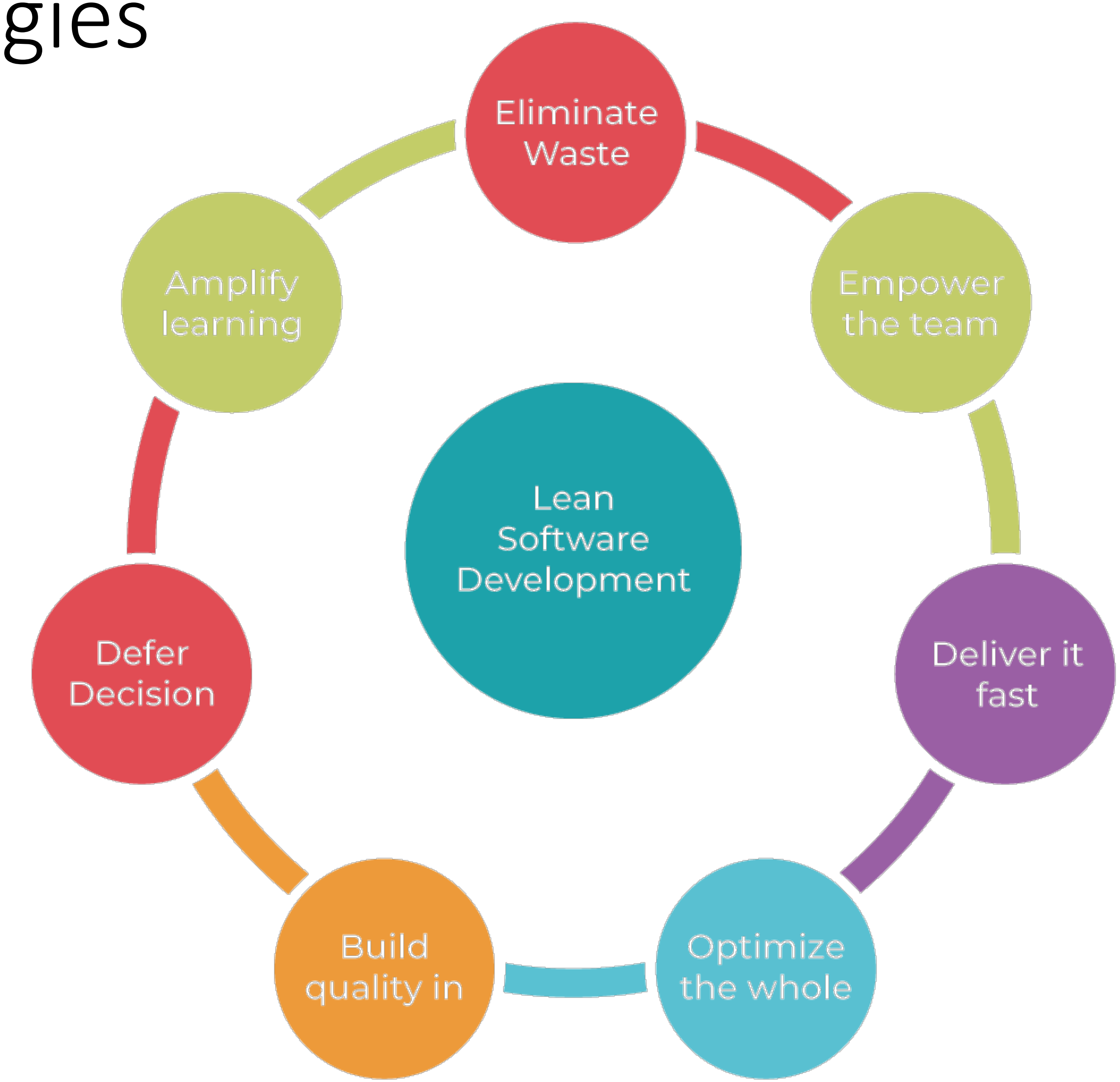
Refactor: Improve code quality

Test Driven Development (TDD)

Highly iterative and incremental development process

Other Agile Methodologies

Kanban: Simplest in IT World;
May Pose time related problems



Some industry-based examples

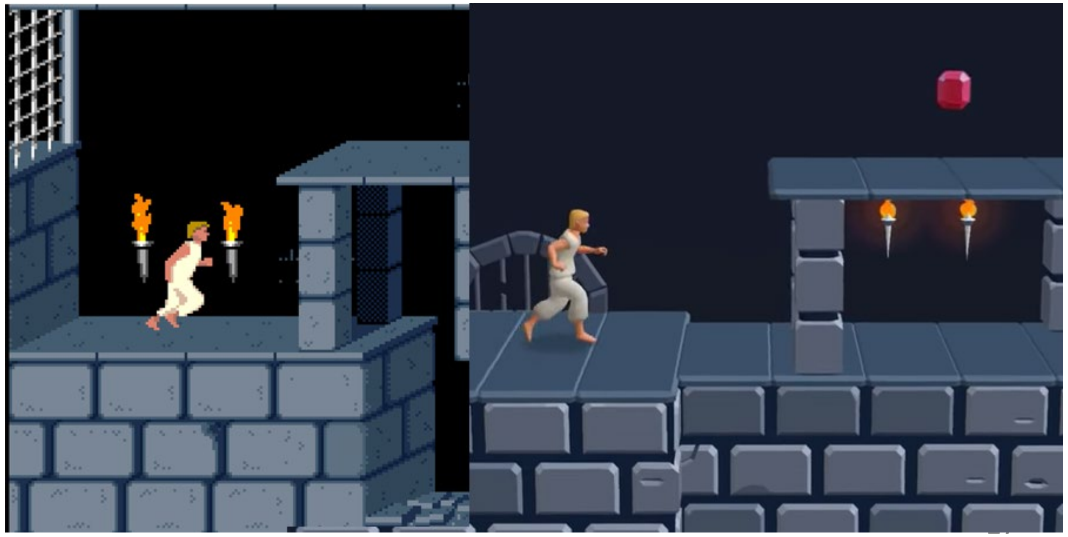
Waterfall

NASA Engineering and Safety Center

Project: The space missions, including the **Apollo moon landings**.

Evolutionary Prototyping

- **Company**: **Broderbund Software**.
- **Project**: The creation of the original **"Prince of Persia"** video game. The initial version of the game was created and then improved upon based on feedback and playtesting.



Some industry-based examples

Spiral

- Early Versions of Windows by Microsoft
- Gantt Chart Software – GanttPRO

Agile

- **Apple, IBM, Microsoft, and Procter & Gamble**
- **Cisco:** defects were reduced by 40% when compared to waterfall
- **Barclays:** 300% increase in throughput
- **Panera Bread:** 25% increase in company sales
- **PlayStation Network:** Saved the company $30 million a year

Choosing the right Software Process Model

Requirements Understanding

Expected Lifetime

Risk

Schedule Constraints

Interaction with Management/Customers

Expertise

As much influence over a project’s success as any other major planning decision

Industry Standards: Factors affecting choice of project LCM

- Degree of Project Complexity
- Work/Time Flexibility
- Project Focus/ Client involvement
- Size of organization
- Role Specialization
- Budget

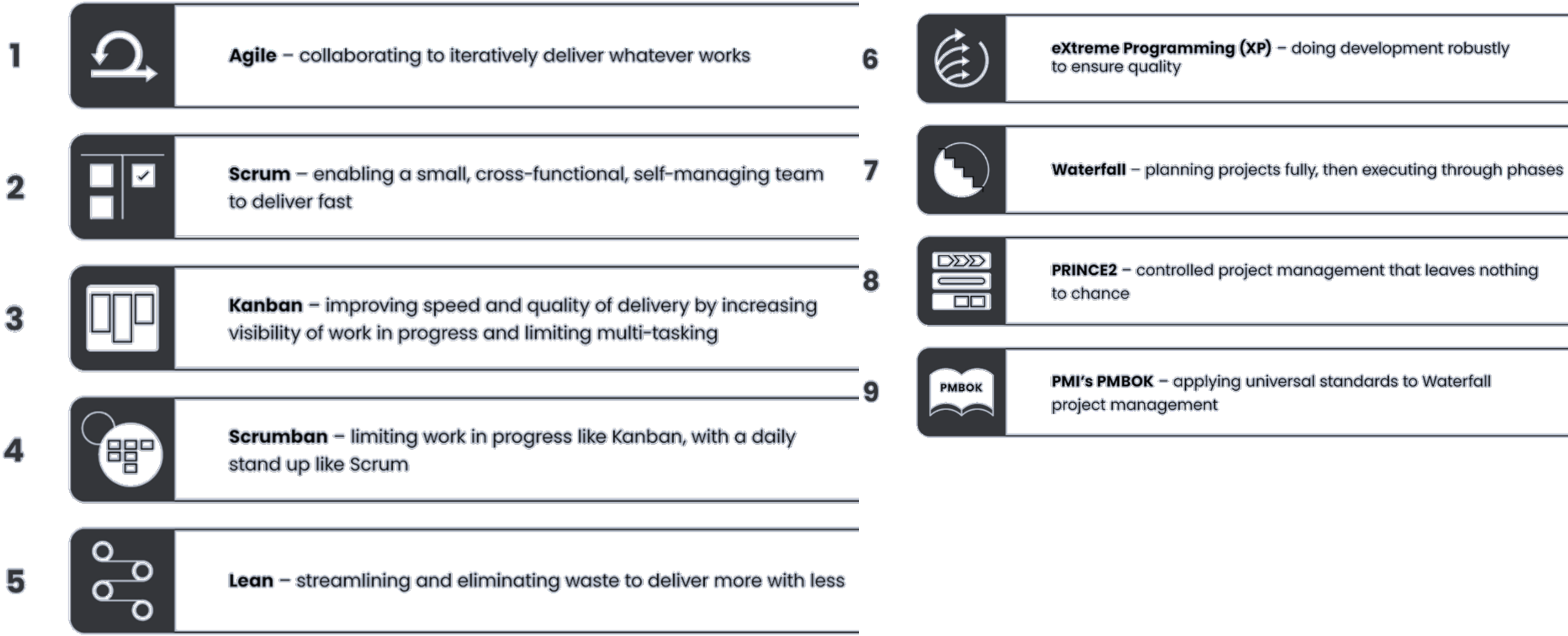
https://asana.com/resources/project-management-methodologies
https://thedigitalprojectmanager.com/projects/pm-methodology/project-management-methodologies-made-simple/

Industry Standards: Factors affecting choice of project LCM

Factors	**Waterfall**	**Evolutionary Prototyping**	**Agile Methodologies**	**Spiral**
Unclear User Requirements	Poor	Good	Excellent	Excellent
Unfamiliar Technology	Poor	Excellent	Poor	Excellent
Complex System	Good	Excellent	Poor	Excellent
Reliable System	Good	Poor	Good	Excellent
Short time schedule	Poor	Good	Excellent	Excellent
Strong Project Management	Excellent	Excellent	Excellent	Excellent
Cost Limitation	Poor	Poor	Excellent	Poor
Visibility of stakeholder	Good	Excellent	Excellent	Excellent
Skills Limitation	Good	Poor	Poor	Poor
Documentation	Excellent	Good	Poor	Good
Component Reusability	Excellent	Poor	Poor	Poor

https://melsatar.blog/2012/03/15/software-development-life-cycle-models-and-methodologies/

Industry Standards: Most Popular Methods



1 **Agile** – collaborating to iteratively deliver whatever works

2 **Scrum** – enabling a small, cross-functional, self-managing team to deliver fast

3 **Kanban** – improving speed and quality of delivery by increasing visibility of work in progress and limiting multi-tasking

4 **Scrumban** – limiting work in progress like Kanban, with a daily stand up like Scrum

5 **Lean** – streamlining and eliminating waste to deliver more with less

6 **eXtreme Programming (XP)** – doing development robustly to ensure quality

7 **Waterfall** – planning projects fully, then executing through phases

8 **PRINCE2** – controlled project management that leaves nothing to chance

9 **PMI's PMBOK** – applying universal standards to Waterfall project management

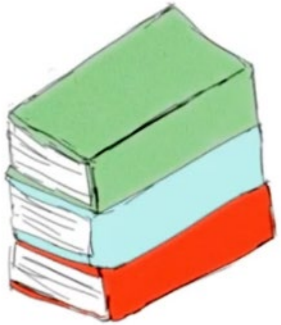
https://thedigitalprojectmanager.com/projects/pm-methodology/project-management-methodologies-made-simple/

Lifecycle Documents

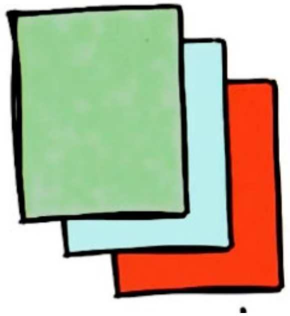
Documenting the activities carried out during the different phases of the lifecycle is a very important task.

Can be used for different purposes like:

- Communicate details of the software systems to different stakeholders
- Ensure the correct implementation of the system
- Facilitate maintenance and so on.



IEEE Documents



Light-weight Documents

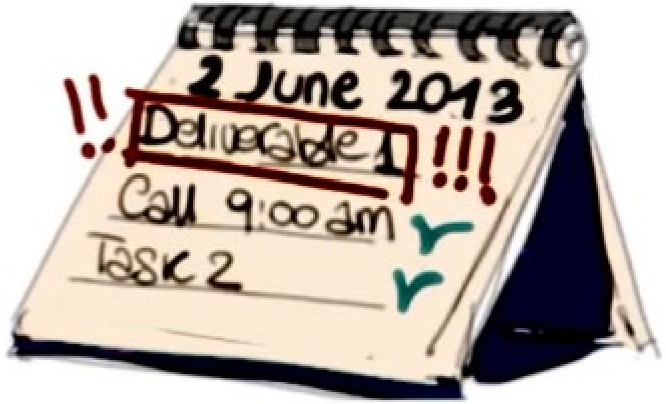
Classic Mistakes : People

Heroics

Work Environment

People Management

Classic Mistakes : Process



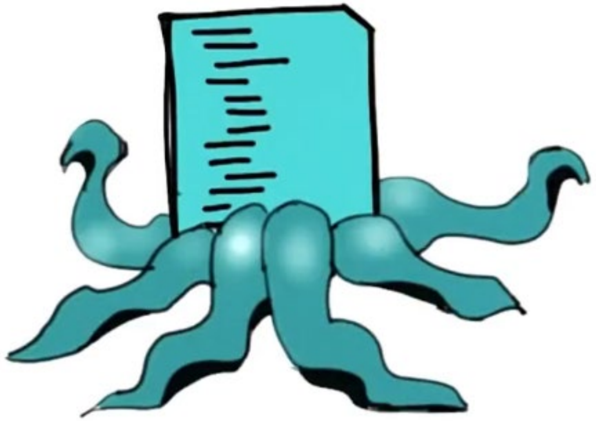
Schedule Issues

Planning Issues

Failure

Classic Mistakes : Product

Gold Plating of Requirements

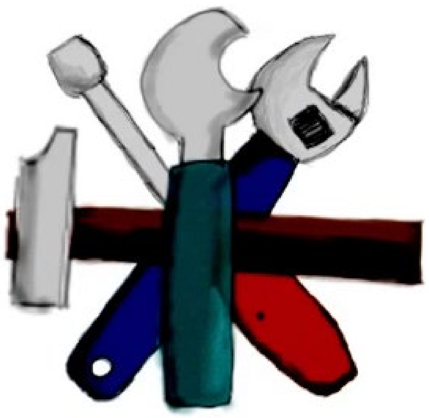


Feature Creep

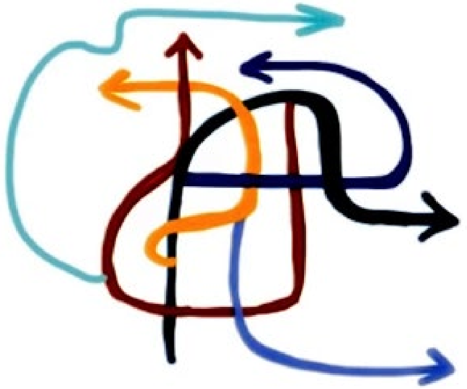
Research ≠ Development

Classic Mistakes : Technology

Silver-Bullet Syndrome



Switching Tools



No version control

Quizizz