SOFTWARE DESIGN
COSC 4353/6353

Dr. Raj Singh

OUTLINE

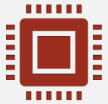
Software Development Process

Software Development Methodologies

SDLC

Agile

SOFTWARE DEVELOPMENT PROCESS

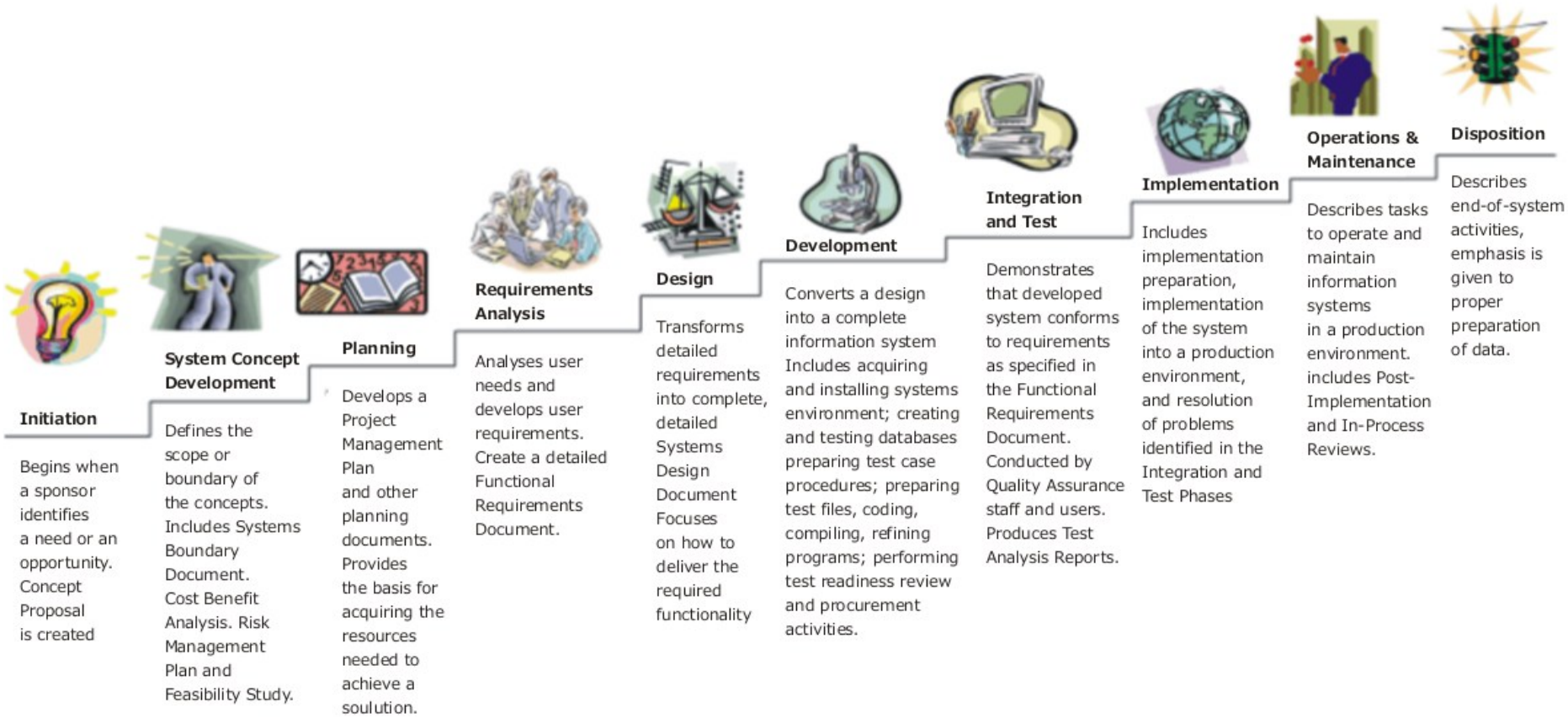


A structure imposed on the development of a software product.

A framework that is used to structure, plan, and control the process of developing an information system.

Several software development approaches have been used since the origin of information technology.

SOFTWARE DEVELOPMENT LIFE CYCLE (SDLC)



SOFTWARE DEVELOPMENT ACTIVITIES

Planning	An objective of each and every activity, where we want to discover things that belong to the project.
Analysis & Design	Analysis of requirements and design of software is done throughout development
Implementation	Implementation is the part of the process where software engineers actually program the code for the project.
Testing	Software testing is the process to ensure that defects are recognized as soon as possible.
Deployment	Deployment starts directly after the code is appropriately tested and approved for release to production environment.
Support	Software training and support is important, as software is only effective if it is used correctly.
Maintenance	Maintaining and enhancing software to new requirements can take substantial time and effort as missed requirements may force redesign of the software.

SOFTWARE DEVELOPMENT MODELS

Prescriptive Models — Traditional

Agile Models — Modern

PRESCRIPTIVE MODELS

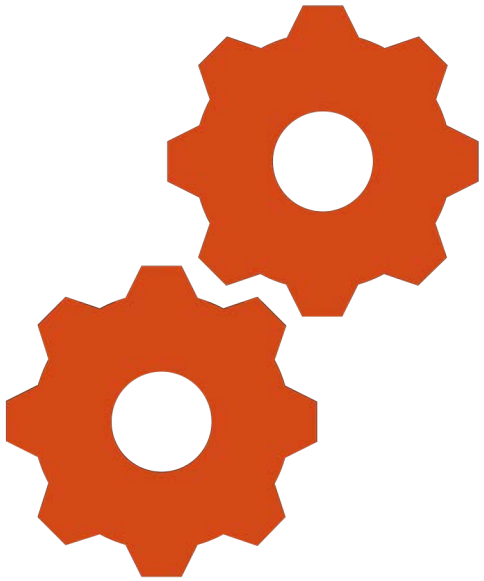
Prescriptive process models advocate an orderly approach to software engineering

That leads to a few questions …

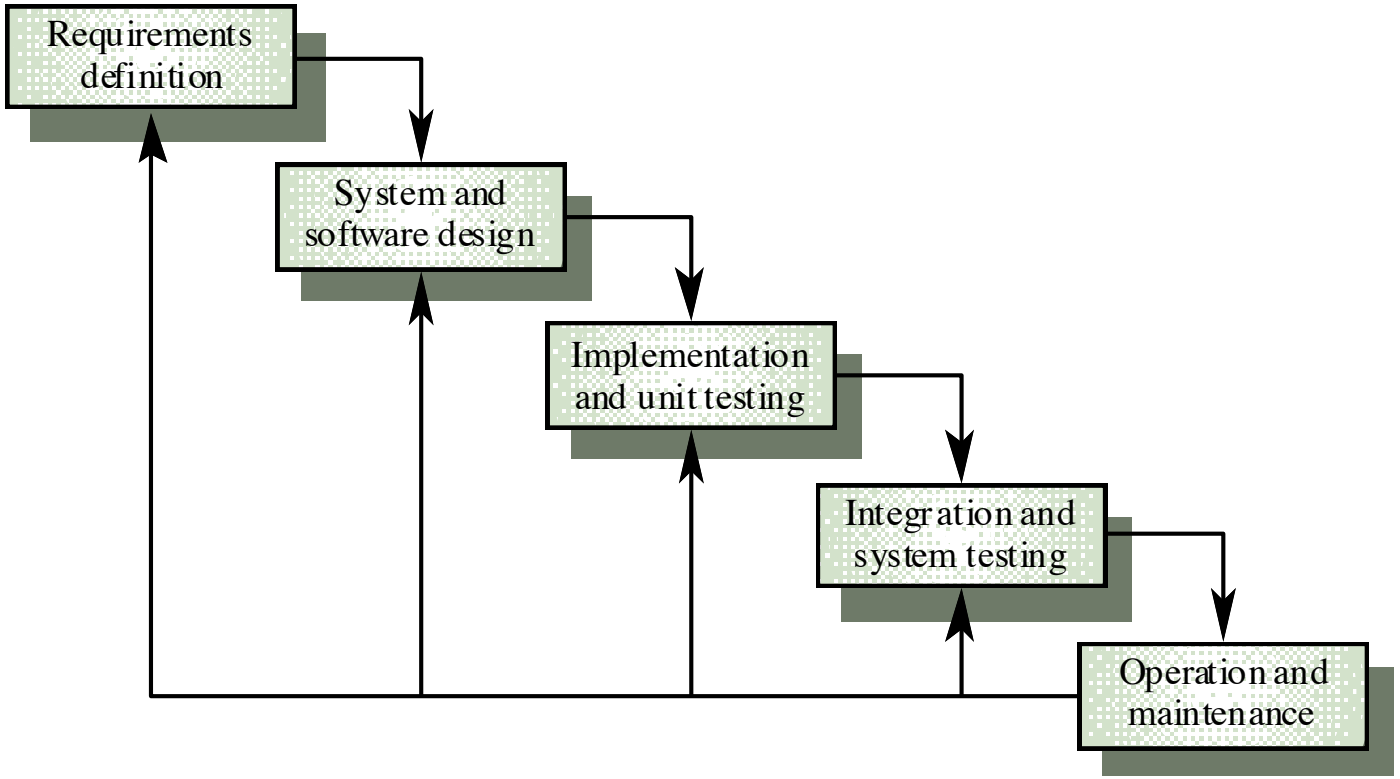
- If prescriptive process models strive for structure and order, are they inappropriate for a software world that thrives on change?
- Yet, if we reject traditional process models (and the order they imply) and replace them with something less structured, do we make it impossible to achieve coordination and coherence in software work?

TRADITIONAL MODELS

- Waterfall
  - a linear framework
- Spiral
  - a combined linear-iterative framework
- Incremental
  - a combined linear-iterative framework or V Model
- Prototyping
  - an iterative framework
- Rapid application development (RAD)
  - an iterative framework



WATERFALL MODEL



WATERFALL MODEL

Software engineers are to follow these phases in order

- Requirements
- Software design
- Implementation
- Testing
- Deployment
- Maintenance

Each phase is dependent on previous step

Next phase starts only if previous step is finished

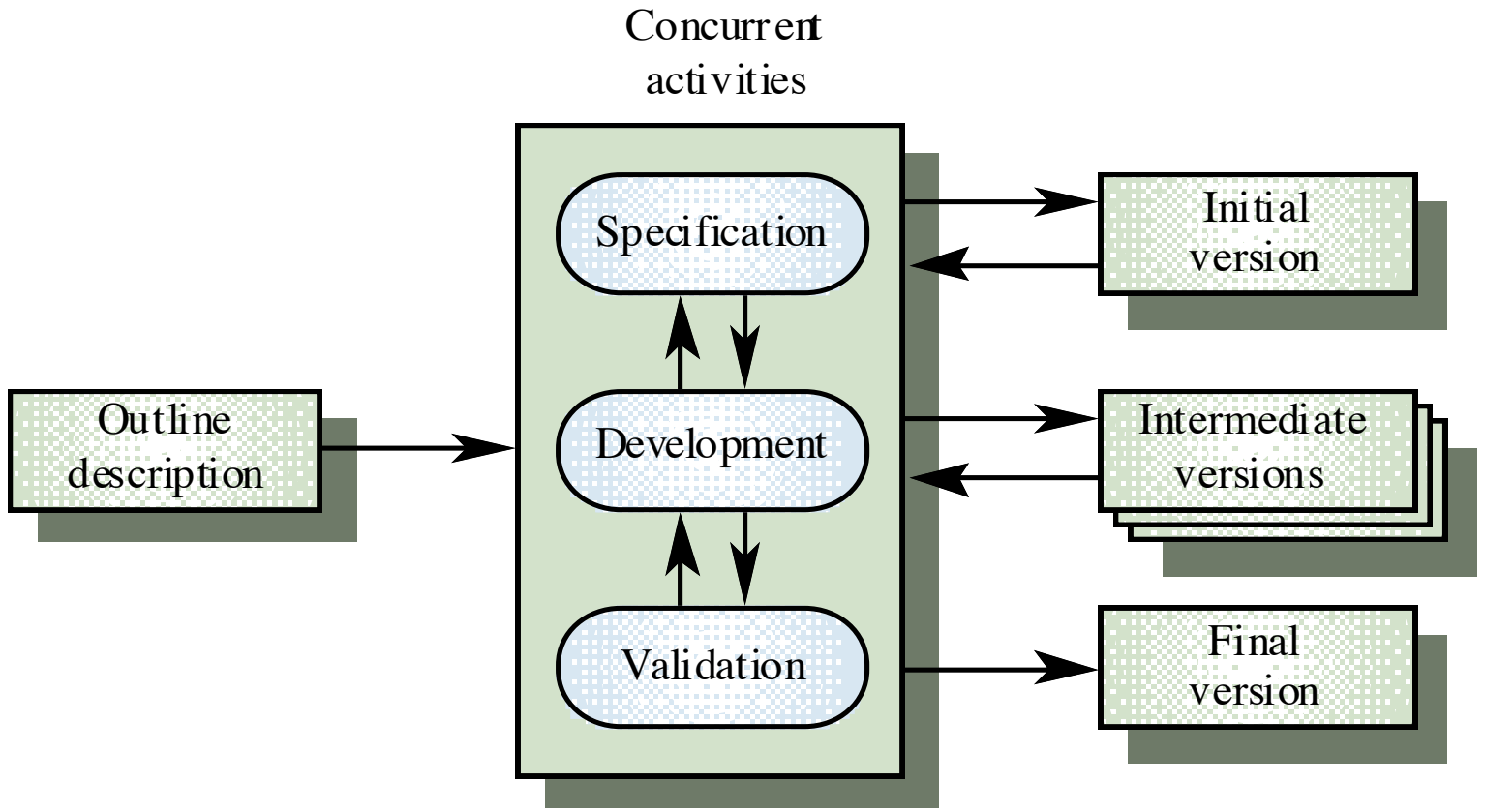
WATERFALL PROCESS CHARACTERISTICS

- Figure out what needs to be done
- Figure out how it will be done
- Then do it
- Verify its done right
- Hand product to customer
- What happens if requirements were not right?

WATERFALL MODEL ISSUES

- Real projects rarely follow the sequential flow that the model proposes.
- At the beginning of most projects requirements are not clear.
- Requirements cannot be changed in the middle.
- The model does not accommodate flexibility very well.
- Development can take very long time and that does not yield a working version of the system until late in the process.

EVOLUTIONARY DEVELOPMENT



EVOLUTIONARY PROCESS CHARACTERISTICS

Modern development processes take evolution as fundamental, and try to provide ways of managing, rather than ignoring, the risk.

Requirements always evolve in the course of a project.

Specification is evolved in conjunction with the software

Not ideal for large systems.

Two (related) process models:

- Incremental development
- Spiral development

EVOLUTIONARY PROCESS CHARACTERISTICS

Modern development processes try to provide ways of managing, rather than ignoring, the risk.

Requirements always evolve in the course of a project.

Specification is evolved in conjunction with the software

Not ideal for large systems.

Two (related) process models:

- Incremental development
- Spiral development

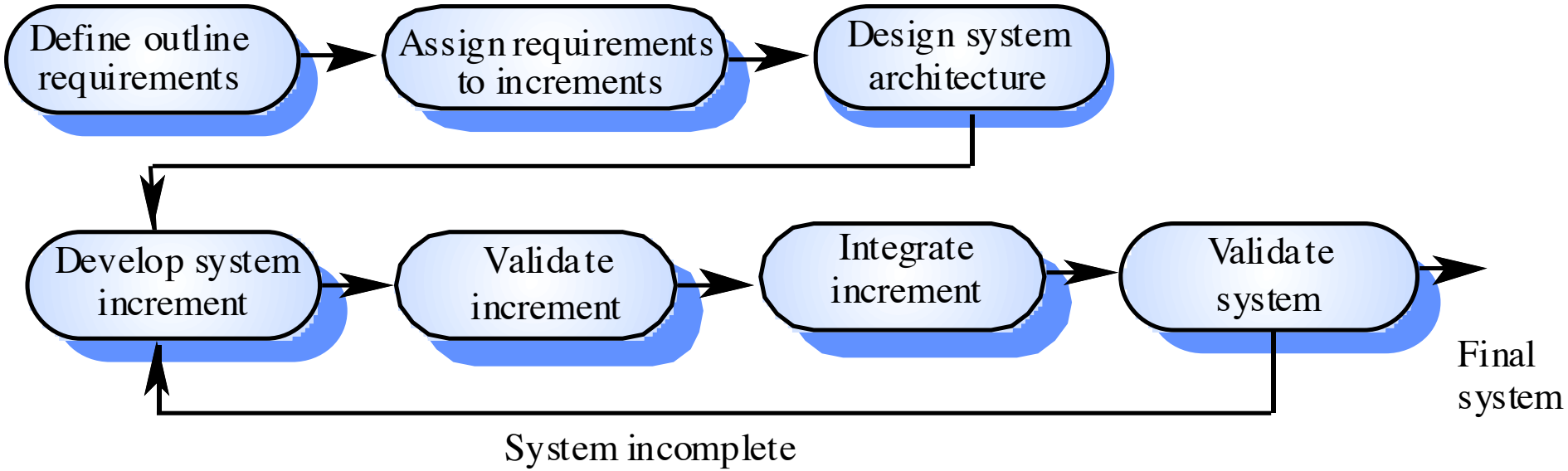
INCREMENTAL DEVELOPMENT

Rather than delivering the system as a single delivery, the development and delivery is broken down into increments with each increment delivering part of the required functionality.

Requirements are prioritised and the highest priority requirements are included in early increments.

Once the development of an increment is started, the requirements are frozen though requirements for later increments can continue to evolve.

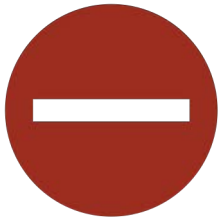
INCREMENTAL DEVELOPMENT



INCREMENTAL DEVELOPMENT ADVANTAGES

- Customer value can be delivered with each increment so system functionality is available earlier.
- Early increments act as a prototype to help elicit requirements for later increments.
- Lower risk of overall project failure.
- The highest priority system services tend to receive the most testing.

INCREMENTAL DEVELOPMENT – PROBLEMS



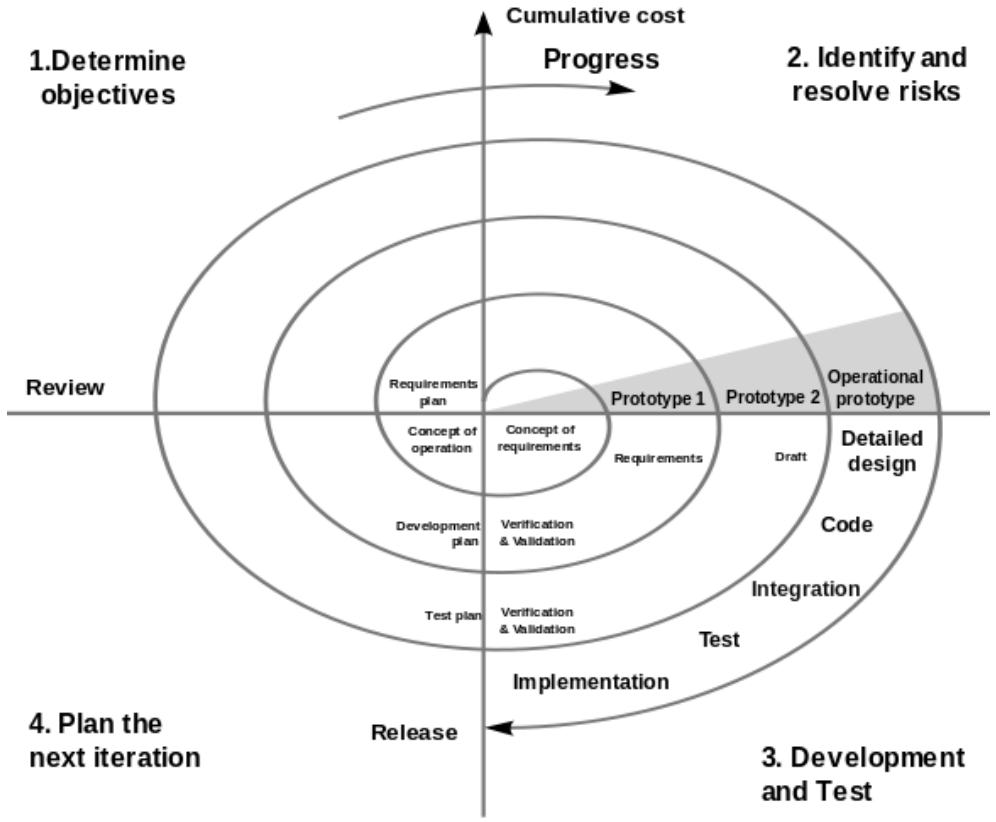
Lack of process visibility.

Systems are often poorly structured.

Not ideal for large systems.

SPIRAL

- The key characteristic of is risk management at regular stages in development cycle
- Combines key aspect of the waterfall model & rapid prototyping
- Good for complex systems.

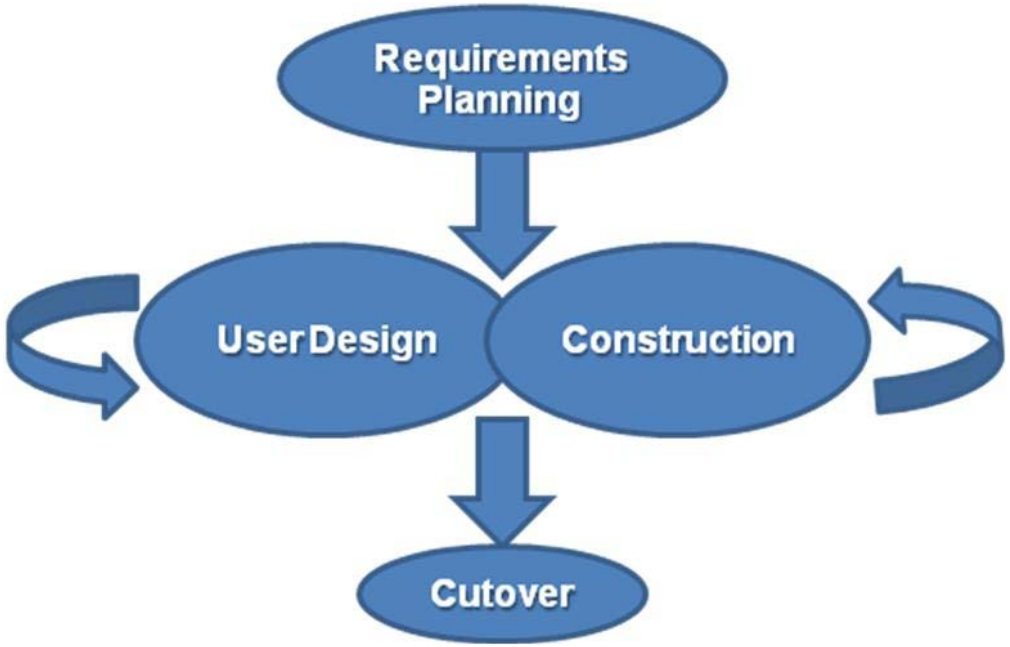


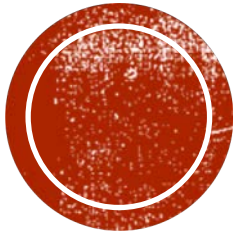
SPIRAL

- Process passing through some number of iterations.
- More emphasis on risk analysis.
- Requires to accept the analysis and act on it.
- Willingness to spend more to fix the issues, which is the reason why this model is often used for large-scale internal software development.
- If the implementation of risk analysis will greatly affect the profits of the project, the spiral model should not be used.

RAPID APPLICATION DEVELOPMENT

- RAD requires minimal planning.
- Faster development.
- Easier to change requirements.
- Iterative & prototyping
- Starts with data models and business process modeling.
- Requirements are verified by prototyping, eventually to refine the data and process models.





AGILE DEVELOPMENT

PROJECT FAILURE – TRIGGER FOR AGILITY

ONE OF THE PRIMARY CAUSES OF PROJECT FAILURE WAS THE EXTENDED PERIOD OF TIME IT TOOK TO DEVELOP A SYSTEM.

COSTS ESCALATED AND REQUIREMENTS CHANGED.

AGILE METHODS INTEND TO DEVELOP SYSTEMS MORE QUICKLY WITH LIMITED TIME SPENT ON ANALYSIS AND DESIGN.

WHAT IS AGILITY?

Effective (rapid and adaptive) response to change

Effective communication among all stakeholders

Drawing the customer onto the team

Organizing a team so that it is in control of the work performed

Yielding …

Rapid, incremental delivery of software

AN AGILE PROCESS

- Is driven by customer descriptions of what is required (scenarios)
- Recognizes that plans are short-lived
- Develops software iteratively with a heavy emphasis on construction activities
- Delivers multiple ‘software increments’
- Adapts as changes occur

AGILE PROCESS

Agile methods are considered

Lightweight

People-based rather than Plan-based

Several agile methods

Extreme Programming (XP) most popular

SCRUM

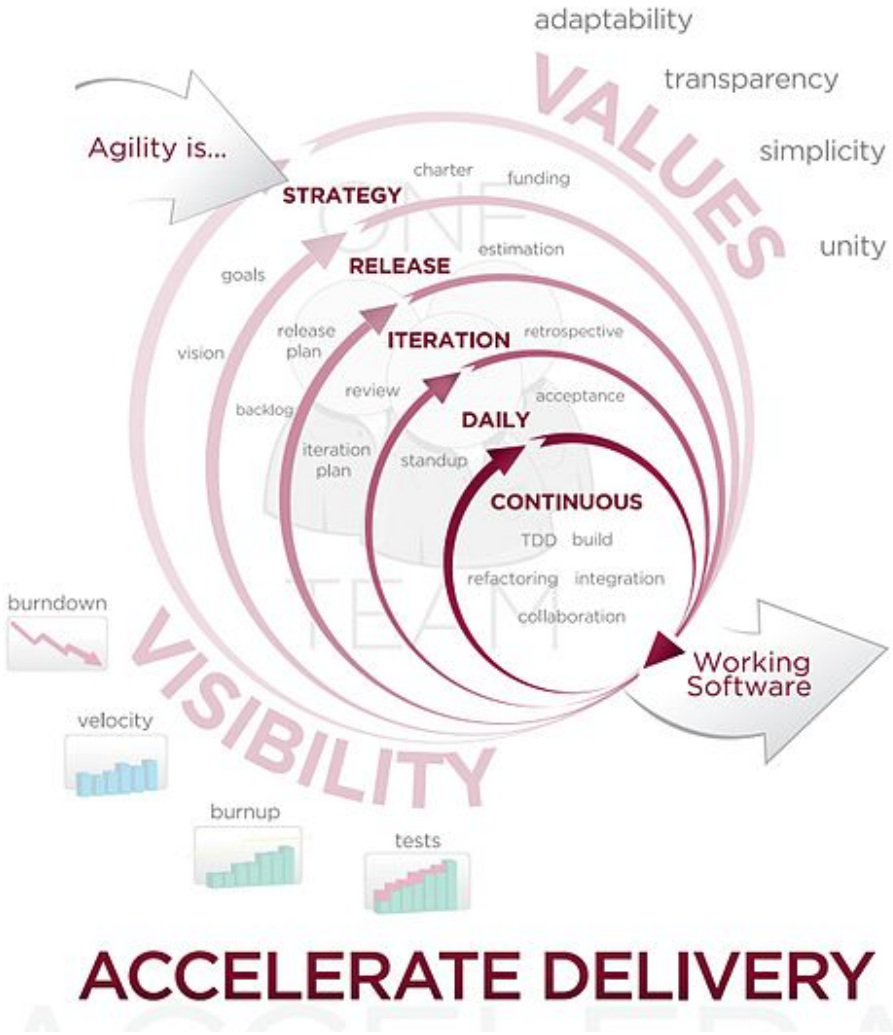
TDD etc…

Agile Manifesto closest to a definition

Set of principles

Developed by Agile Alliance

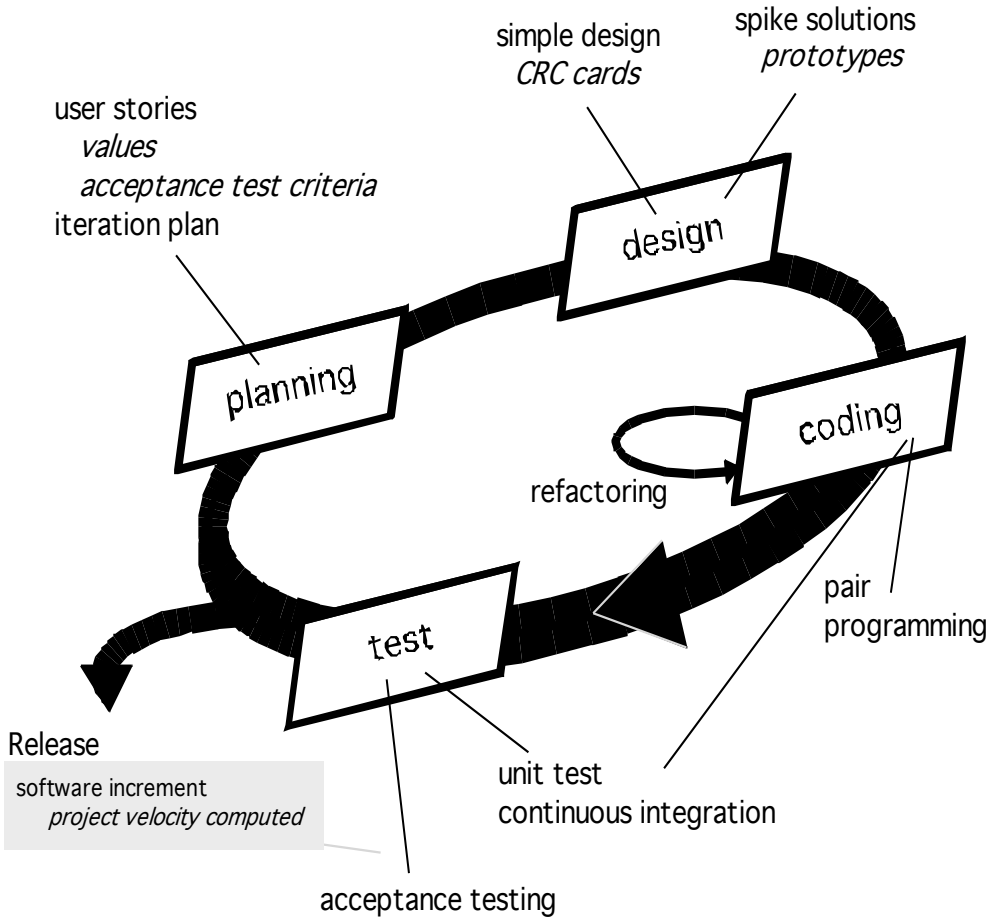
AGILE DEVELOPMENT



EXTREME PROGRAMMING (XP)

- Follows agile process
- The phases are carried out in extremely small (or "continuous")
- First write automated tests as concrete goal for development
- Then coding. Complete only if all tests passed
- Design and architecture emerge out of refactoring
- The incomplete but functional system is deployed or demonstrated
- Move to next part of the system

EXTREME PROGRAMMING (XP)



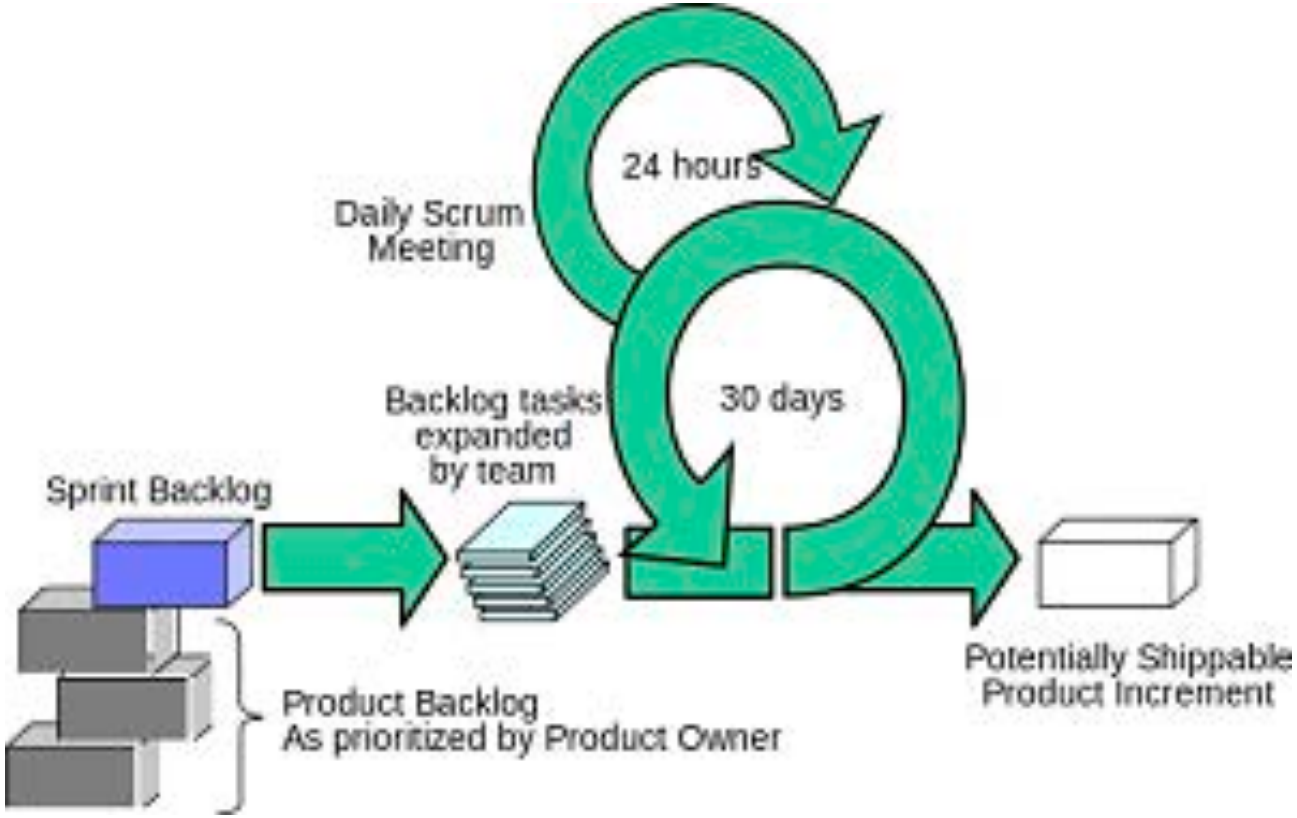
SCRUM

- Scrum is a framework for agile software development
- Enables the creation of self-organizing teams by encouraging co-location of all team members
- Testing and documentation are on-going as the product is constructed
- Work occurs in “sprints” and is derived from a “backlog” of existing requirements
- Meetings are very short and sometimes conducted without chairs
- “demos” are delivered to the customer with the time-box allocated

SCRUM TERMS

Scrum Team	product owner, development team, scrum master
Sprint	Timeboxed iteration of a continuous development cycle
Planning	Work and effort necessary to meet their **sprint** commitment
Product Backlog	List of all things that needs to be done within the project
Sprint Backlog	list of all things that needs to be done within a sprint
Daily Meeting	15-minute meeting to provide status update
Review	**Review** of the team's activities during the **Sprint**
Retrospective	What went well and continue? What can be improved? Actions

SCRUM – PROCESS FLOW



TEST DRIVEN DEVELOPMENT

A process that relies on the repetition of a very short development cycle

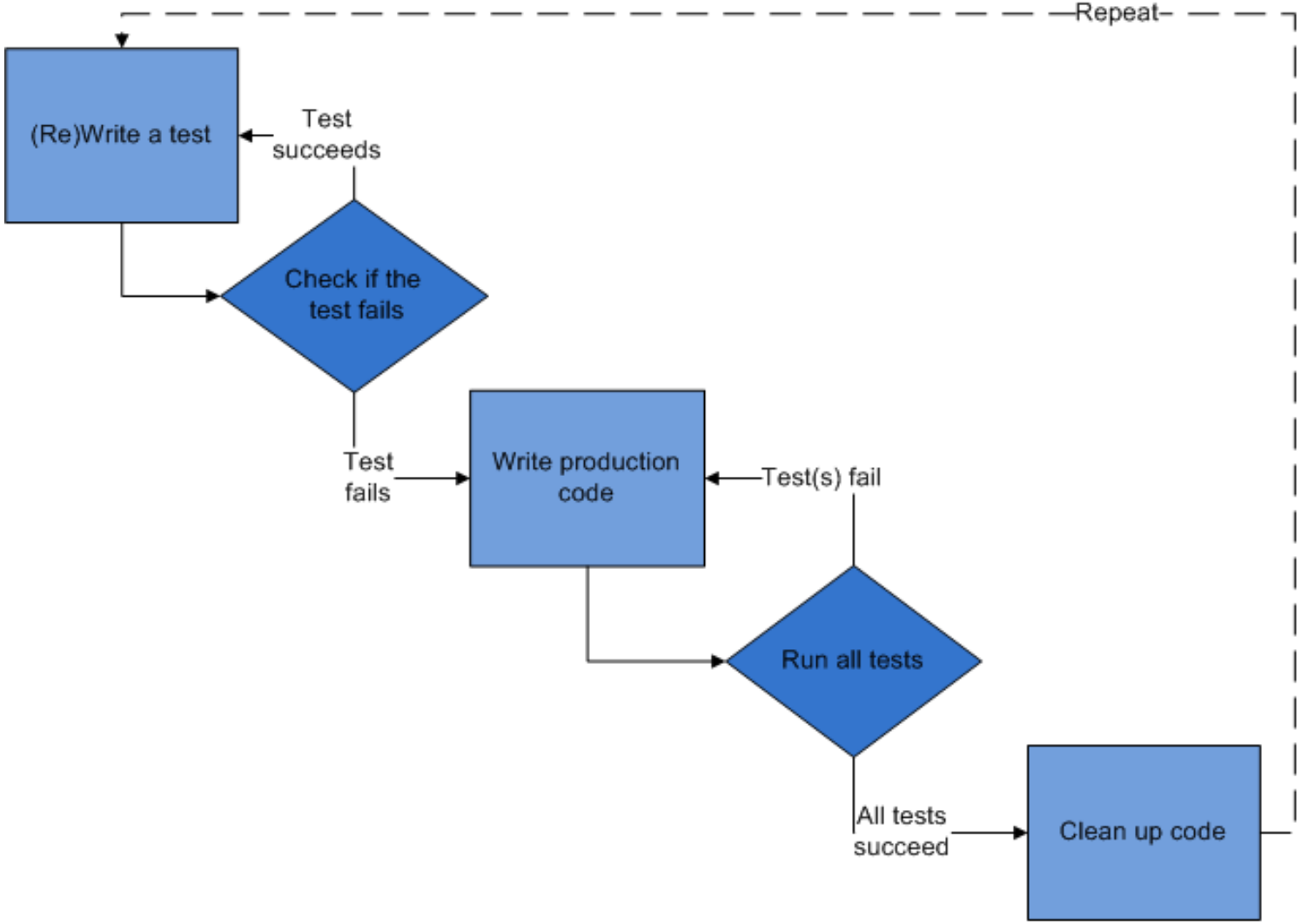
Based on test first programming concept of XP

First write an (initially failing) automated test case that defines a desired improvement or new function

Write minimum amount of code to pass the test

Finally re-factor the code to acceptable standards

TDD


Repeat
(Re)Write a test
Test succeeds
Check if the test fails
Test fails
Write production code
Test(s) fail
Run all tests
All tests succeed
Clean up code

HOMEWORK

Review class notes.

Additional reading: latest trends in software development.

Start a discussion on Google Groups to clarify your doubts.