Indian Institute of Science
Bangalore, India
भारतीय विज्ञान संस्थान
बंगलौर, भारत

DS286 | 2016-08-17

L3: Software Design

Yogesh Simmhan

simmhan@cds.iisc.ac.in

CDS
The Department of Computational and Data Science

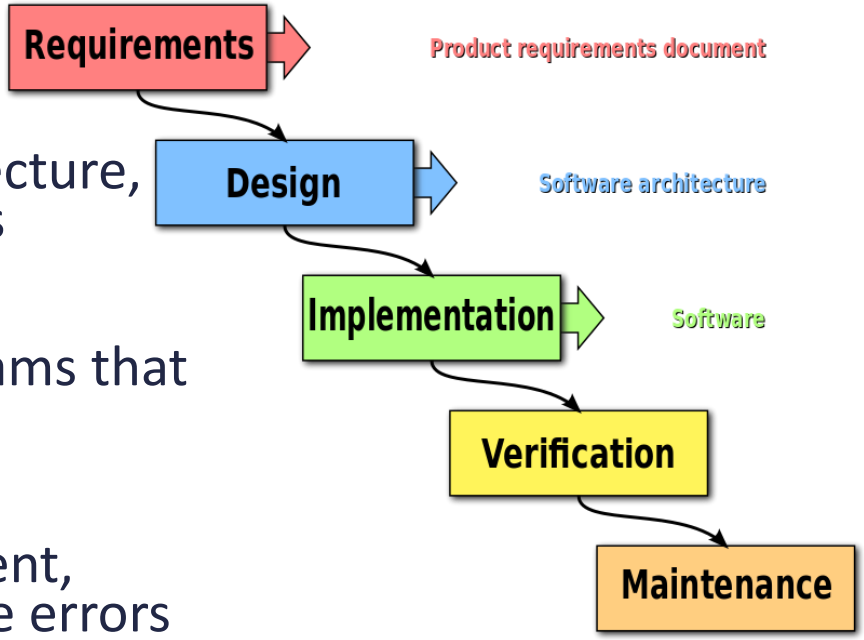
Software Engineering

- Product vs. Process
  - *Software* is the product that you "develop"
  - *Software engineering* is the process used to develop the software
- Software is different from manufacturing
  - Design is everything!
- Software, unlike hardware, does not wear out
  - But changes/patches cause them to deteriorate…
- Software is often custom-built, rather than assembled
  - Seems easier than reusing…but has consequences on manageability

Software engineering: A practitioner's Approach, Pressman, 1997

Linear Sequential Model…**Waterfall**

- Analysis
  - Functional, performance, information, interfacing requirements
- Design
  - Data structure, software architecture, interface definitions, algorithms
- Code
  - Follow design to develop programs that meet the requirements
- Test
  - Assurances on program statement, functionality, interfacing, handle errors
- Maintenance
  - Fix errors, adapt to architectural/organizational change



Software engineering: A practitioner's Approach, Pressman, 1997
https://en.wikipedia.org/wiki/Software_development_process

Waterfall: Pro's & Con's

Advantages

- Easy to explain to the user· Structures approach.
- Stages and activities are well defined
- Helps to plan and schedule the project
- Verification at each stage ensures early detection of errors / misunderstanding
- Each phase has specific deliverables

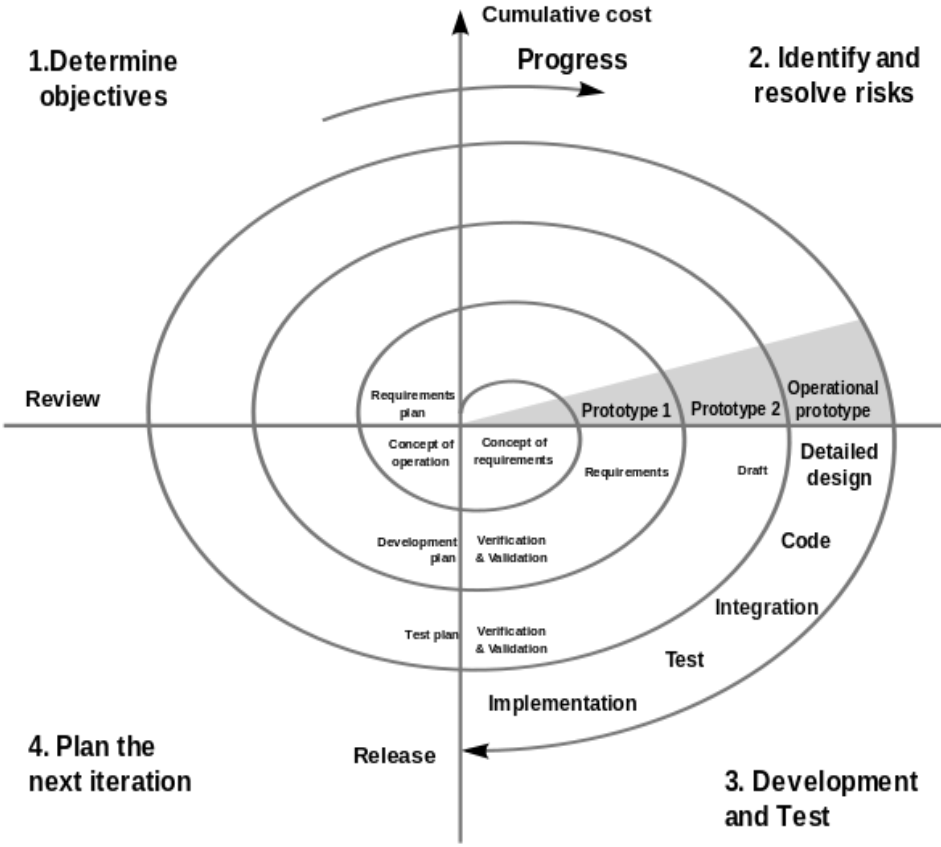
Disadvantages

- Assumes that the requirements can be frozen
- Very difficult to go back to any stage after it finished.
- Little flexibility and adjusting scope is difficult and expensive.
- Costly and required more time, in addition to detailed plan

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

Spiral Model

- Communication between customers, developers
- Planning tasks, timelines, resources
- Risk analysis: technical & management
- Detailed design & engineering
- Develop, test, install, document, train
- Release, Evaluate



Software engineering: A practitioner's Approach, Pressman, 1997
https://en.wikipedia.org/wiki/Software_development_process

Spiral: Pro's & Con's

Advantages

- Estimates of budget, schedule become more realistic as work progresses, because important issues are discovered earlier.
- Early involvement of developers
- Manages risks and develops system into phases

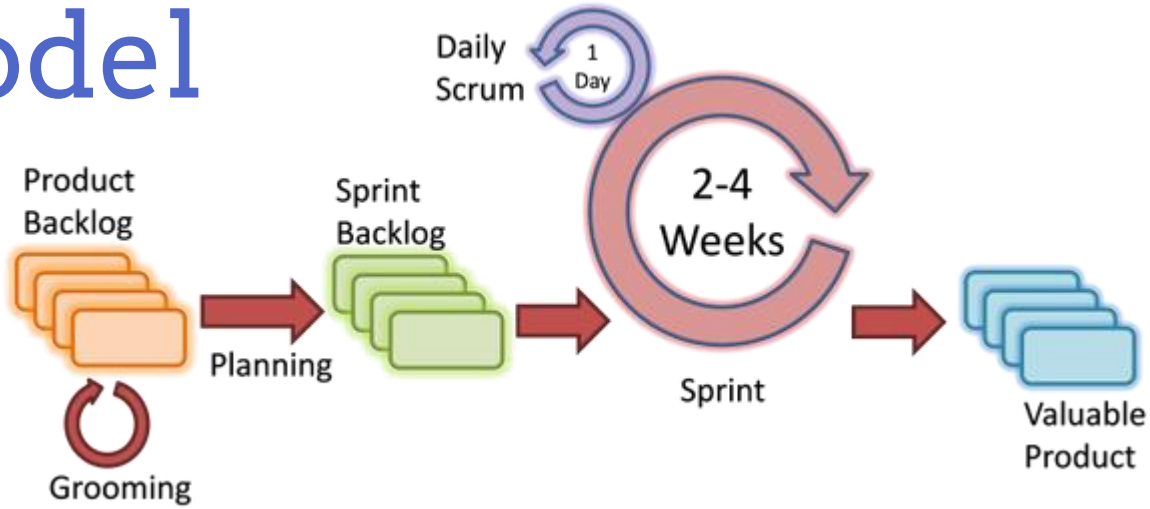
Disadvantages

- High cost and time to reach the final product.
- Needs special skills to evaluate the risks and assumptions
- Highly customized limiting re-usability

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

Agile Model

Agile Manifesto Principles



- Iterative, Incremental and evolutionary
  - Short timelines, integrated teams, demo to end users
  - Working software is necessary at each iteration!
- Face to face communication with end users
- Very short feedback loop, adaptation cycle
  - Daily “scrum” stand-up meetings
- Continuous integration & testing, design patterns, pair-programming, refactoring

https://melsatar.wordpress.com/2012/03/15/software-development-life-cycle-models-and-methodologies/
https://en.wikipedia.org/wiki/Software_development_process

Agile: Pro's & Con's

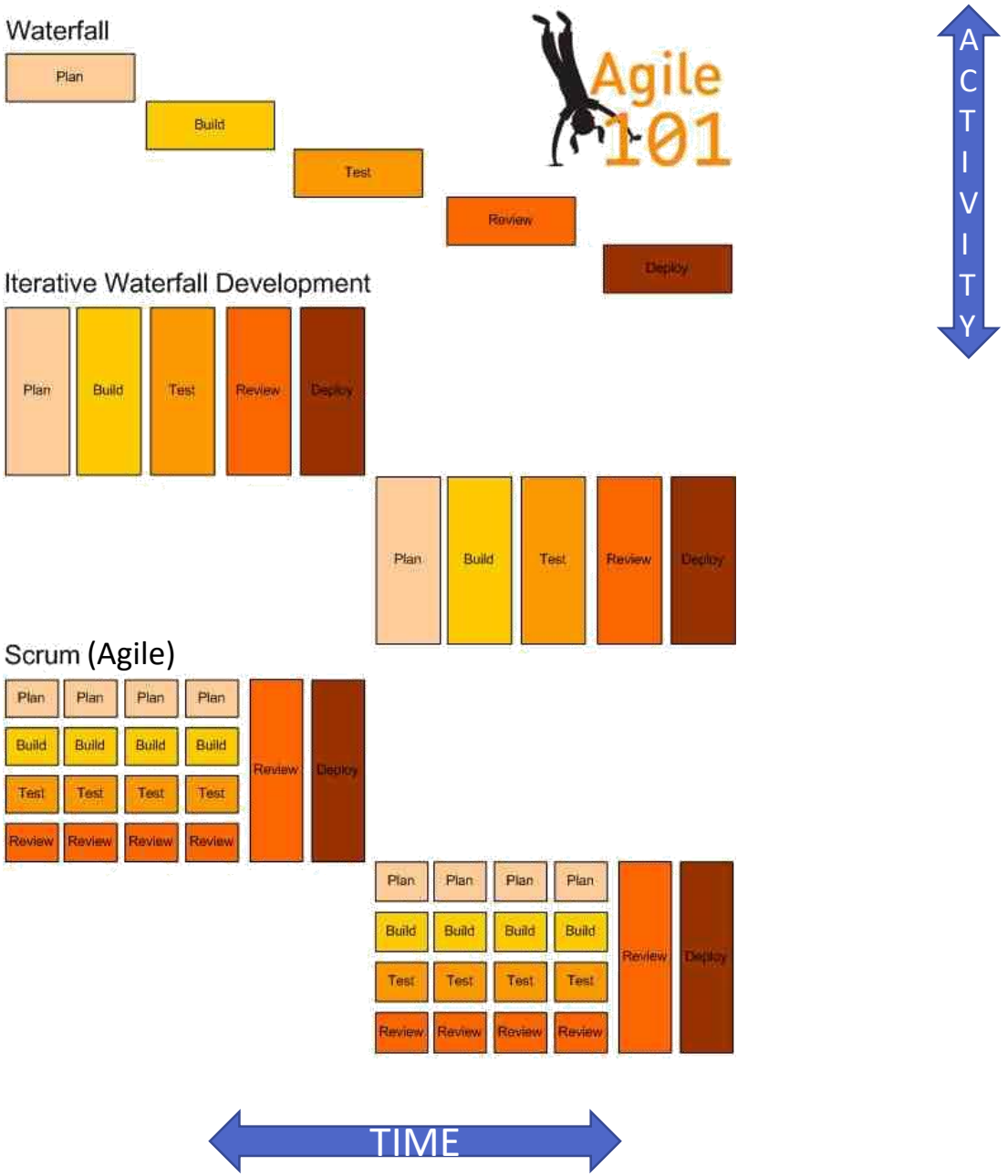
Advantages

- Decrease the time required to avail some system features.
- Face to face communication and continuous inputs from customer representative leaves no space for guesswork.
- The end result is the high quality software in least possible time duration and satisfied customer

Disadvantages

- Scalability· Skill of the software developers
- Ability of customer to express user needs
- Documentation is done at later stages
- Reduce the usability of components.
- Needs special skills for the team.

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



http://www.allaboutagile.com/agile-vs-waterfall-vs-iterative-vs-lean-software-development-in-pictures/

Testing

- Quality assurance for developed code
  - Gives the correct result for correct inputs
  - Handles the errors for incorrect inputs
- White-box and Black-box testing
  - Look into the internals vs. Look at the interfaces
- Unit testing: Individual components
- UI testing: Of user interfaces (dummies!)
- Bounds testing: Min/Max values, out of range
- Integration testing: Interfacing between components, across systems

Documentation & Coding Style

- Comments
  - Classes, Functions, Variables

```
/**
 * Class header comments
 */
// Inline comments
```

- Coding Style
  - Variable, Constant & function names
    - Use meaningful names!
    - `int num_errors; // "num" is a widespread convention.`
    - `AddTableEntry()`
    - `PI_VALUE`
- Braces, White spaces
  - Align open and close braces, indent using spaces/tabs

http://www.edparrish.net/common/cppdoc.html
https://google.github.io/styleguide/cppguide.html#General_Naming_Rules
https://www.stack.nl/~dimitri/doxygen/manual/docblocks.html

Announcements & Tasks

- Self study
  - Read Sahni Textbook: Chapter 1 "C++ Review"
  - Complete Chapter 1 exercises 4, 6, 10, 13, 16, 19, testing of program 1.36.
- Sign up for mailing list, if not done so already
  - `mailman.serc.iisc.in/mailman/listinfo/ds286.aug16`
- Fill in Google Sheets with details
  - Link posted on mailing list
- Finish Assignment 0 by Fri Aug 19 *(0 points)*
  - ***Submission instruction???***
- Assignment 1 will be posted by Aug 19, due Aug 28 *(50 points)*
- '`turing`' Cluster access will be provided by Fri Aug 19
  - All submissions MUST work (compile, run) on cluster node!

Questions?