Software Engineering

Lectures :
Tuesday 08.00-10.00 am
Thursday 08.00 -10.00 am
By Alex Tumwesigye
atumwesigye@gmail.com, +256774149775

Assessment

- Two tests: 10% each

CAT 1:??

CAT 2:??

- One project:10%
- Home works: 10% (can be transferred to CATS)
- Exam: 60%
- Attendance at least 75%
- NB: Any exam/test malpractices, consider yourself not to have sat for that test.

Objectives of the course

- Equip students with the ability to apply software engineering principles and methods of design in the development and maintenance of Telecommunication engineering applications

Content

- Software engineering basics and definition
- Phases of software development
- Software methodologies
- Software development teams
- Introduction to c++ programming
- Functions
- Pointers and arrays
- Memory (dynamic and static memory)
- OOP - Classes and interfaces, abstraction, inheritance, polymorphism, etc
- Software engineering applications in Telecommunication Engineering (Project assignment)

Software engineering

- Application of engineering principles and methods of design in the production of software

➢ Covers technical aspects of build software systems

➢ Managerial issues e.g directing programming teams, scheduling, planning and budgeting

Phases of software development

- Analysis – identify the problem, gather requirements from clients, etc
- Design – The layout, etc. Use of requirements to clarify the picture of the software
- Implementation – put the design to action – coding (programming)
- Testing – test the code and debug to correct errors, maintenance can also be done

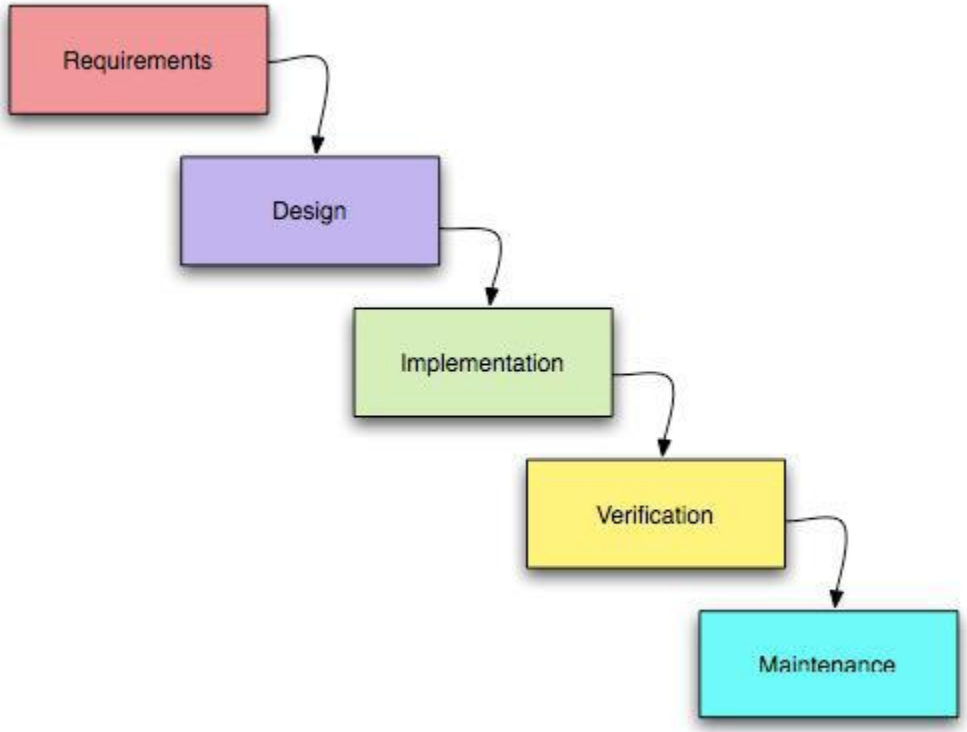
Software methodologies

- Waterfall

- Project is segmented into hierarchy of chunks, phases.
- Analysis comes first, then design,implementation and testing
- Phases are executed sequentially

Analysis

Waterfall


Requirements
Design
Implementation
Verification
Maintenance

Good concepts of waterfall

- Work is done in stages
- Content reviews are conducted between
- stages, and
- Places emphasis on documentation

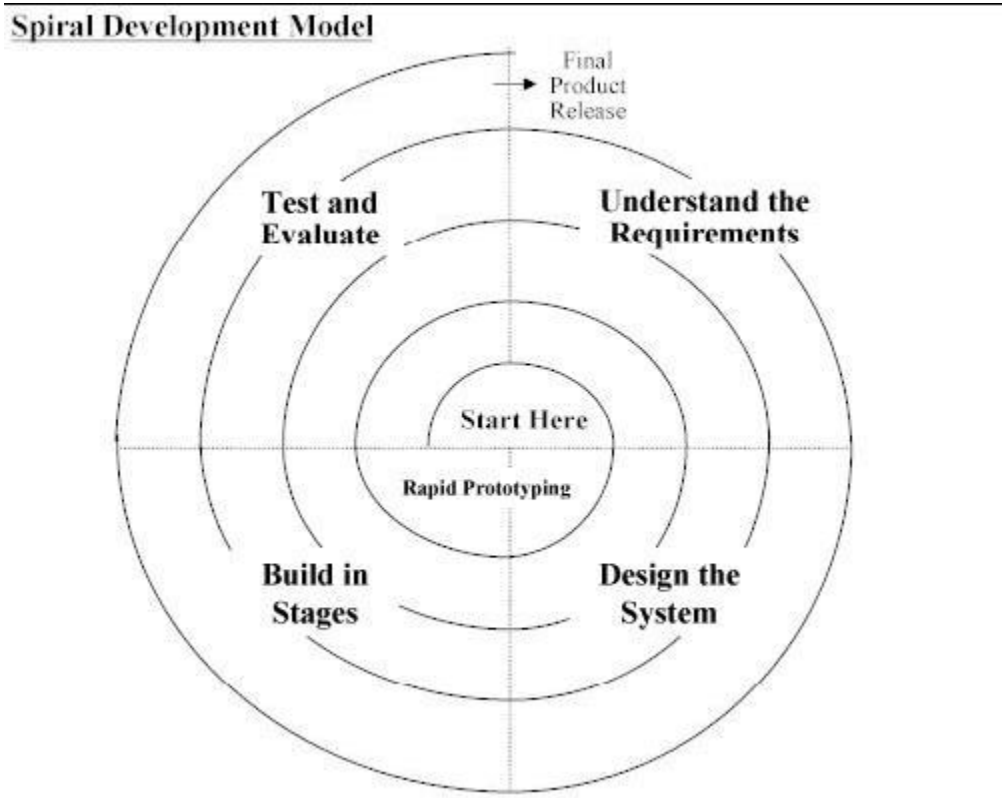
Critics of waterfall

- Impossible to get one phase of the project development perfected
- Software development projects must be open to external factors – not the case
- Only part of the team are qualified for a certain phase hence “code monkeys – people who programs for a living”

Spiral methodology

- Evolution of the waterfall method
- Projects are both incremental and interactive
- Teams start small and benefit from enlightened trial and error along the way
- Chosen for large, extensive complicated projects
- Introduces the concept of prototyping

Spiral methodology



Good concepts of spiral

- System is able to grow sequentially since it allows feedback in the system development life cycle
- Avoids stagnation of the system at a certain phase

Critics of spiral

- No control over the oscillations
- When will the solution be finally reached?

Prototyping

- Building experimental systems in an explanatory way
- Advantages

➢ May provide proof of concept

➢ Give users an idea of how the final system will look like

➢ Enables higher output for user

➢ Cost effective

Prototyping

- Advantages (cont'd)

➢ Encourages active participation from users and producers

➢ Increase system development speed

Prototyping

- Disadvantages

- User expectation on the prototype may be above its performance
- May cause systems to be unfinished
- May cause systems to be implemented before they are finished
- Difficult to manage projects
- Not suitable for large applications

Software team organization

- A team is a cooperative unit.
- A software engineering team is a group of people drawn from diverse but related groups, assigned to perform a well defined function for an organization or project.
- Can part-time participants or can have other primary responsibilities

Building teams

- Depends on
  - People available
  - Needs of the project
  - Needs of the organization
- Effective teams consists of people fulfilling a variety of roles. Each person can take one or more roles

Project roles

- Analayst
- Programmer
- Project manager

Homework: Describe other roles other than the above which can be assigned to team members in a software engineering project

Due: Next lecture

Team organization

- Horizontal approach – Consists of specialists who take on a specific tasks/role and only that role
- Vertical approach – consists of generalists who take on a variety of roles
- Hybrid approach – a combination of specialists and generalists

Horizontal approach

- Teams work on several use cases simultaneously, each member working on their own aspects of the use case
- Good communication required between members
- Defined processes and quality gates for specialists are needed
- Advantages
  - High quality of work on each aspect
  - External groups e.g users, operations staff, interact with a small group of specialists who understand their exact needs.

Horizontal approach

- Disadvantages
  - Specialists do not appreciate the work of other specialists, thus there is disconnect in various aspects of the project.
  - Information required by the “back-end” people may not be gathered by the “front-end” people.
  - Project management is difficult because of competing priorities, visions & needs of specialists.

Vertical approach

- Use cases are assigned to small groups, who implement the different use cases.
- Good communication required to avoid implementation of common functionalities
- Advantages
  - Developers gain a wide range of skills
  - Smooth end to end development on an individual use case basis.

Vertical approach

- Disadvantages
  - Generalists are hard to obtain – high paid consultants
  - Generalists do not have special technical expertise required to quickly solve detailed problems
  - Subject matter experts may have to work with several groups of developers, increasing their burden
  - All generalists are not equal!

Review next lecture

- Programming languages (c, c++, PHP/MySQL, Java, perl, Python,etc)
- Project 1: Implement a scheduling system for lectures for different departments in the Faculty depending on the available lecture rooms using c or c++ programming, indicate the objectives, justification, scope, methodology, etc. The solution must include functions, classes, inheritance, etc.
- Deadline: 2 weeks before the Final exam