SE笔记

Software Engineering

1. Introduction

Software engineering is an engineering approach on a software development of systematic application.

1.1 Software Development Process

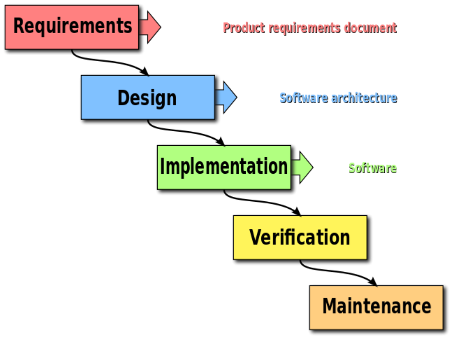
In software engineering, a software development process is the process of dividing software development work into smaller, parallel or sequential steps or subprocesses to improve design, product management. It is also known as a software development life cycle (SDLC).

Methodologies of software development process : Most modern development processes can be vaguely described as agile. Other methodologies include waterfall, incremental development, rapid application development.

- Waterfall development

  The waterfall model is a sequential development approach, in which development is seen as flowing steadily downwards (like a waterfall) through several phases, typically:

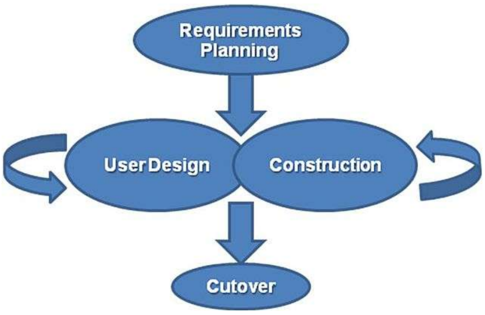
  - Requirements analysis resulting in a software requirements specification.
  - System Design
  - Implementation
  - Testing
  - Integration
  - Deployment
  - Maintenance

- Rapid application development (RAD)

  RAD favors iterative development and the rapid construction of prototypes instead of large amounts of up-front planning. The lack of extensive pre-planning generally allows software to be written much faster, and makes it easier to change requirements.

  Specifically, it is to quickly develop a prototype for customers to use, and then develop and iterate based on customer feedback... The advantage of RAD is that the final software will be more in line with customer expectations.



- Incremental development
  Various methods are acceptable for combining linear and iterative systems development methodologies, with the primary objective of each being to reduce inherent project risk by breaking a project into smaller segments and providing more ease-of-change during the development process.
- Agile development
  Agile software development uses iterative development as a basis but advocates a lighter and more people-centric viewpoint than traditional approaches. Agile processes fundamentally incorporate iteration and the continuous feedback that it provides to successively refine and deliver a software system.
  In software development, agile practices include requirements discovery and solutions improvement through the collaborative effort of self–organizing and cross–functional teams with their customer(s)/end user(s), adaptive planning, evolutionary development, early delivery, continual improvement, and flexible responses to changes in requirements, capacity, and understanding of the problems to be solved. Other elements of agile include: doing extensive code review, unit testing of all code, a flat management structure, code simplicity and clarity.

2. SDLC

2.1 Software Requirements

The requirement for a system are the description of what the system should do, we usually start with four questions to understand software requirement: **Who, Where, What, How**–
who did what, where and how.

- What should we do
  - Identify the problem to be solved: Understand the requirements through multiple interviews
  - Clarify the scope of requirements
  - Estimated cost
  - Risk assessment
- Documentation and tools
  - BRD: Business Requirement Document
  - URD: User Requirement Document
- UML
  - Use case diagram
  - Activity diagram

2.2 Software Design

Software design usually involves problem-solving and planning a software solution. This includes both a low-level component and algorithm design and a high-level, architecture design.

- Generally including:
  - System architecture design
  - Data structure design
  - Prototype design
  - UI design
  - Test case development
- Documentation and tools
  - PRD (Product Requirement Document)
  - Axure (for prototype design)
  - Photoshop, Sketch (for UI)
  - Test cases document
  - Database design document
  - Development plan
- UML
  - Class diagram
  - Object diagram
  - Sequence diagram
  - Deployment diagram

2.3 Software Construction

Software construction is the main activity of software engineering, and the main work of the team at this stage is to develop system functions.

- What does the development team do:
  - Front end development
  - Back end development
  - Database construction
  - Interface development
  - Algorithm development
- Documentation and tools
  - Develop document
  - Jira (for project management and team collaboration)

2.4 Software Testing

Software testing is an empirical, technical investigation conducted to provide stakeholders with information about the quality of the product or service under test, with different approaches such as unit testing and integration testing. It is one aspect of software quality. As a separate phase in software development, it is typically performed by quality assurance staff or a developer other than the one who wrote the code.

- What does the development team do:
  - unit testing
  - integration testing
  - functional testing
  - Acceptance test
  - Pressure test
- Documentation and tools

    - Test document
    - Acceptance document

2.5 Software Maintenance

Software maintenance is modifying and updating software applications after distribution to correct faults and to improve its performance.

- What does the maintenance team do:
    - Check if the system is functioning properly
    - Handling exceptions
    - Deploy updates
- Documentation and tools
    - Operation and Maintenance Manual
    - Deployment document

Reference:

Software Engineering – Wikipedia