

SCUT 2019 Fall Software Engineering Review

qixuan

December 11, 2019

Abstract

Chapter 1: Overview
Chapter 2: Software Processes
Chapter 3: UML modeling
Chapter 4: Ruby
Chapter 5: PetriNet
Chapter 6: Requirement Engineering
Chapter 7: Formal Specification
Chapter 8: Architectural Design
Chapter 9: Rails_MVC
Chapter 10: Design Pattern
Chapter 11: Rails_Test
Chapter 12: Software Testing
Chapter 13: Basic Path

Chapter 1: Overview

Software = Program + Data + Document

Software crisis(1968 at a conference)

new computer hardware

larger and more complex software

software cost more than predicted

New techniques and methods needed to control the large software systems

Software Engineering - born on a conference in German, 1968.

A **Engineering Principle** to handle Software

Three basic elements of SE:

•**Method:** refers to a formal procedure; a formal “recipe” for accomplishing a goal that is typically independent of the tools used

•**Tool:** an instrument or automated system for accomplishing something in a better way

•**Procedure:** a combination of tools and techniques to produce a product

Methods of SE:

•**UML and other models**

•**Design Pattern**

•**Formal method**

•**Agile Development**

Procedure of SE:

•**Analysis(Requirement specification)**

•**Design**

•**Implementation**

•**Testing**

Environment and Tools:

•**Ruby**

•**Rails**

Other topics:

•**Declarative Programming: Prolog**

•**Distributed Architecture: CORBA**

Analysis: decompose a large problem into smaller, understandable pieces (abstraction is the key) **Synthesis:** build (compose) a software from smaller building blocks (composition is challenging)

Changes in SE: Desktop Computing, Object Technology, Problems with waterfall, Time to market, Networking, User interfaces, Shifts in economics.

Software Lifecycle

Software Process

software process model is a simplified description of a software process, include activities of software process, software products and the role of people involved.

To model software development: Waterfall approach, Iterative development, Component-based software engineering.

Development process models

Although software engineering involves all three processes, in this chapter we discuss only the development process, which is shown outside the cycle . The development process in the software lifecycle involves four phases: **analysis (requirement specification), design, implementation and testing**. There are several models for the development process. We discuss the two most common here: the **waterfall model** and the **iterative model**.

The waterfall model

In this model, the development process flows in only one direction. This means that a phase cannot be started until the previous phase is completed

- **Analysis (Requirement specification)**
- **Design**
- **Implementation**
- **Testing**

The iterative model

In the iterative model, software is developed in a series of steps.

Each increment is a entire waterfall process.

Software Development Steps:

Requirement Analysis and Definition: procedure-oriented or object-oriented, UML, output a specification document.

System Design: All part of system defined, Structure charts, Modularity(**Coupling** between modules in a software system must be minimized and **Cohesion** within modules in a software system must be maximized).

Program Design

Program Implementation: Choice of language, Software quality.

Unit Testing: glass-box(white box), black box(exhaustive, random, boundary-value), **basis path testing** is a method in which each statement in the software is executed at least once.

Integration Testing

System Testing

System Delivery

Maintenance

Documentation

For software to be used properly and maintained efficiently, documentation is needed. Usually, three separate sets of documentation are prepared for software: **user documentation, system documentation and technical documentation**.

Documentation is an ongoing process.