RUET

EDited by: rmshoeb\_cse\_16

Software Engineering

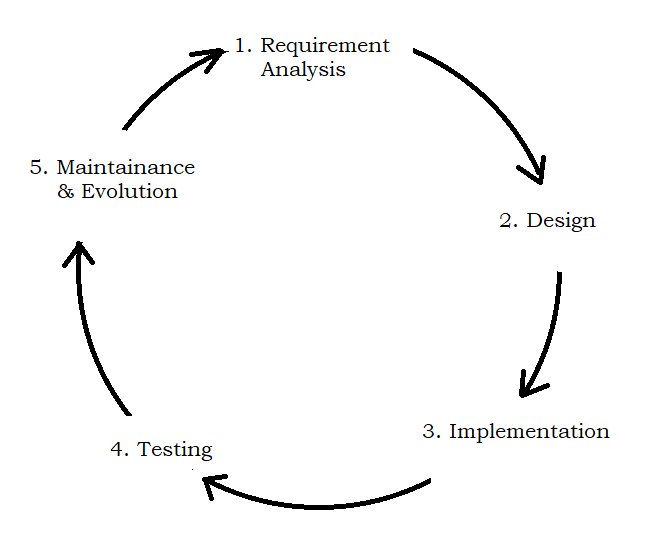
CSE 3105 by SSG and MAR

Introduction

**Introduction to Software and its nature**

* What is a software?
* Software characteristics
* Types of software
* System software
* Application software
* Software development process: 3 phases

1. Definition
2. Development
   * Design
   * Implementation
   * Testing



Software Development Life Cycle (SDLC)

1. Support

* Correction
* Adoption
* Enhancement
* Prevention

**Professional and Ethical Responsibility of a Software Engineer**

* Like other engineering disciplines, software engineering is carried out within a social and legal framework that limits the freedom of people working in that area. As a software engineer, you must accept that your job involves wider responsibilities than simply the application of technical skills. You must also behave in an ethical and morally responsible way if you are to be respected as a professional engineer.

It goes without saying that you should uphold normal standards of honesty and integrity. You should not use your skills and abilities to behave in a dishonest way or in a way that will bring disrepute to the software engineering profession. However, there are areas where standards of acceptable behavior are not bound by laws but by the more tenuous notion of professional responsibility. Some of these are:

1. *Confidentiality* You should normally respect the confidentiality of your employers or clients irrespective of whether or not a formal confidentiality agreement has been signed.
2. *Competence* You should not misrepresent your level of competence. You should not knowingly accept work that is outside your competence.
3. *Intellectual property rights* You should be aware of local laws governing the use of intellectual property such as patents and copyright. You should be careful to ensure that the intellectual property of employers and clients is protected.
4. *Computer misuse* You should not use your technical skills to misuse other people’s computers. Computer misuse ranges from relatively trivial (game playing on an employer’s machine, say) to extremely serious (dissemination of viruses or other malware).

To know more, read from “Software Engineering, Sommerville – ninth edition”

Article - 1.2: Software engineering ethics

What is Software Engineering?

* Software engineering is an engineering discipline that is concerned with all aspects of software production.

Software Process Model

**Different types of Software process model and their implementations**

* A software process model is a simplified representation of a software process.
* Waterfall model
* Advantages
* Disadvantages
* Prototype model
* Rapid Application Development (RAD)
* Incremental model
* Spiral model
* Project backlog

User story

Sprint

Sprint backlog

* Scrum

Scrum method

* Agile process model
* Feature Driven Development (FDD): Scrum

Advantages

Disadvantages

* Test Driven Development (TDD): Extreme Programming

Advantages

Disadvantages

* Difference between these two:
* Difference between iterative and incremental type models:

Iterative model is an approach where thefocus is put on creating a working prototype first and adding features in development cycles where the Incremental Development stages (Waterfall model) are done for every cycle. Each increment in the incremental approach builds a complete feature of the software, while in iterative, it builds small portions of all the features.

**Factors in Choosing a Software Process**

* Customer involvement
* Stable requirements
* Team size / proximity
* Developer experience
* Familiarity with technology
* Familiarity with domain
* Severity of impact of incorrect analysis
* Anticipated changes in technology

**Costs of Software Engineering**

Software Requirement Analysis

**Software requirement analysis and their applications**

**Software prototyping**

**Basic concept of different formal specification**

Design of Software

**Software design and its different techniques**

**Software configuration managements**

**System structuring**

**Control model**

**Modular decomposition**

**Domain-specific Architecture**

Software Testing

**Software validation and verification**

<https://www.geeksforgeeks.org/software-testing-basics/>

* **Verification and validation planning**
* **Software’s testing strategies and different type of testing techniques**

Regression testing

* Retest all
* Featurewise test: notun kono feature add korle ta test kora
* Priority wise: ekta software a kichu feature thake ja must kaj kortei hobe. Oigula test kora.

Top-down-approach

Bottom-up-approach

Cluster

Smoke test: software ekebare kaj e korena emon kichu ache kin a ta test kora

Show stopper

Validation testing

* Alpha test
* Beta test

System testing

* Recovery test
* Security test
* Stress test
* Performance test

Condition testing (kind of white box testing)

Data flow testing

Loop testing

Concatenated loop

* **Art of debugging**

Software Quality Assurance

* Software configuration method

**Management and its quality assurance**

**Software cognitive fundamentals**

**Concepts of Software Re-engineering and Web engineering**

Advanced Topics

**Software reliability matrix**

**Software reliability specification**

**Statistical testing and Reliability growth modeling**

**Use of CASE tools and Technological support in Engineering Software**

**Introduction to Unified Modeling Language (UML)**

Testing: 2 types (according to book)/ 3 types (in real life)

* Black box testing
* White box testing
* Grey box testing

Software testing

* Objectives-3
* Good test case
* Successful test case

Software development process

* System Engineering
* Requirement analysis
* Design
* Implementation
* Unit testing
* Integration testing
* Validation testing
* System testing

Software testing principles

* All tests should be traceable to customer requirements.
* Tests should be planned before implementation.
* Pereto principle
* Testing should begin “in the small” and progress towards “in the large”.
* Third party tester

Testability

* Operability
* Observability
* Controllability
* Simplicity
* Stability
* Understandability

Checkout:

* NVC Framework
* Model view control

Estimation

* Size oriented matrices
* FP estimation
* FP oriented matrix
* Difference

Project management and concept

When is testing over?

* Never, the burden just shifts to end-user.

Strategic issues

* Build robust product that test itself.

Peer review