<table>
<thead>
<tr>
<th>Course</th>
<th>Basic Software Engineering</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trainer</td>
<td>Hau Tran - Harry</td>
</tr>
<tr>
<td>Designed by</td>
<td>EPG</td>
</tr>
<tr>
<td>Last updated</td>
<td>1-Sep-2008</td>
</tr>
</tbody>
</table>
Contents

• Concepts
• Introduction
• Requirement
• Design
• Implementation
• Testing
Course Objectives

After this course, attendees can:

- Understand the general software engineering process
- Capable to analyze requirements and design architecture which conforms to the requirements
- Capable to implement the software in the software units, integrate them and perform testing on the software
Concept

- **Work product (WP):** any artifact produced by a process. Ex: source code, documents, parts of the product, processes... Ex: SRS, SDS

- **Work item (WI):** A work item is a record of a portion of the work on a project that must be fulfilled or evaluated to complete the project. Ex: internal code review, coding feature A

- **Software Requirement Specification (SRS):** A document which is developed to describe the customer’s requirements in details.

- **Software Design Specification (SDS):** A document which is developed to describe how the system will be built, as well as what tools and what techniques are to be used to build it **in High Level.**

- **Detail Design Specification (DDS):** Same to SDS, **but in Detail Level.**

- **Requirement Traceability Matrix (RTM):** This matrix defines the mapping between customer requirements-design structures-implementation item & test cases in order to trace/track if any of them is failed which impacts to requirements.
Project Manager (PM): The individual who is responsible for the planning, coordination and controlling of a project.

Business Analyst (BA): Is responsible for analyzing the business needs of their clients and stakeholders to help identify business problems and propose solutions.

Customer (CUS): An individual or organization that purchases and receives a product, or service provided by the organization.

Technical Lead (TL): Leader in all technical issues.

Software Engineer (SE): Team members who develop product.

Quality Control (QC): Team members who are responsible for product quality.
Questions

- What is quality processes?
- Why do we need these processes?
The process used to develop software is often known as Software Development Life Cycle (SDLC). There are many different versions of SDLC.

We will use greatly simplified version of the SDLC consisting of 6 basic processes as figure 1. For the purpose of this course, we would discuss about 4 development processes (Requirement, Design, Implementation & Testing).

The work-products delivered by each phase are shown arranged within each phase as figure 2.
Introduction (cont)

Figure 1

Diagram showing the process flow:
1. Begin
2. Requirement
3. Design
4. Implementation
5. Testing
6. Deployment
7. Maintenance
8. End

Figure 1
Introduction (cont)

Figure 2
Requirement

- The main purpose of requirement is to establish common understanding & agreement between the customer & project team on what software will do
- Customer, Project Manager, Technical Lead and Business Analysis involve in this process
- The requirements document brings the following benefits:
  - Customers can see early on if their needs will be met
  - Developers can estimate the effort involved in creating the application
  - Project manager has a basis for a project plan
  - Quality Control people have a basis for testing the application
Requirement Workflow

1. Raw Requirements
2. Requirements Repository
3. Approved RTM
4. Approved SRS

1. SRS
2. Requirements Repository

1. SRS
2. RTM

1. Change Request
2. Updated SRS
3. Updated RTM

1. Requirements Repository
2. Requirements Repository

Project Manager
Technical Lead
Business Analysis

Technical Lead
Business Analysis

Customer

Project Manager
Business Analysis

Project Manager
Technical Lead

Elicit Requirements

Analyze and Specify Requirements

Validate Requirements

Manage Requirement Changes

End
Elicit Requirements

- PM gathers high level expectation of software and raw requirements which come from CUS via meeting minute, mail and document them into Requirement Repository.
- PM & TL & BA elicit requirement based on techniques as Interviews, Scenario, Prototype, Observation (Refer to Requirement Guideline).
- PM sends Requirement Repository to CUS for review.
Analyze & Specify Requirements

- TL and BA classify requirements from Requirements Repository. Requirements can be classified on a number of dimensions, ex: Functional or Non-Functional, Priority and Scope.
- TL develops conceptual models (Use Case model) which comprise models of entities to reflect their real-world relationships and dependencies.
- All these info are documented in SRS.
- Bidirectional relationship of requirements and SRS are documented in RTM.
- What does SRS address?
  - Functionality: What is the software supposed to do?
  - External interfaces: How does the software interact with people, the system’s hardware, other hardware, and other software?
  - Non-functionality: Performance, security, quality attributes.
Validate Requirements

PM, BA & CUS validate requirements by some of following methods:

- Peer review
- Prototyping

Refer to Requirement Guideline

- PM reviews and approves SRS
- CUS reviews and approves Requirement List in Requirement Repository
Manage Requirement Changes

- When there is a requirement change request, PM analyzes the impact of the changed/new requirement on existing commitments.
- PM approves/rejects Requirement Changes
- TL updates Requirement Repository, RTM and SRS
- PM reviews Requirements Repository and RTM to identify inconsistencies between the requirements and the related plans.
In this section, we have learnt about:

- **Elicit Requirements activity**: is to gather information and develop customer/product-component requirements.
- **Analyze & Specify Requirements activity**: is to ensure that each requirement (functional and non-functional) is analyzed and traceable, complete and correct reflection of the customer’s needs and documents them in SRS.
- **Validate Requirements activity**: is to seek out and correct problems before committing to implement the requirements.
- **Manage Requirement Changes activity**: to identify inconsistencies between the requirements and the related plans.
Design is defined as the process of defining the architecture that realizes the software requirements.

The purpose of design is to develop the architecture and to prepare test plan & integration test case for implementation.

Involved Role
- Project Manager (PM)
- Business Analyst (BA)
- Technical Lead (TL)
- Software Engineer (SE)
- Quality Control (QC)
Design
**Design Workflow**

1. **Begin**
   - Technical Lead
   - Software Engineer
   - Quality Control
   - Project Manager
   - Technical Lead

2. **Develop Software Design**
   - 1. SRS
   - 2. RTM
   - 1. initial SDS (High Level Design) or DDS (Detail Design)
   - 2. updated RTM

3. **Define Test Plan**
   - 1. approved SDS or DDS
   - 2. Test Plan
   - 3. Test Cases

4. **Review Design & Test Plan**
   - 1. SDS or DDS
   - 2. Test Plan
   - 3. Test Cases

5. **End**
   - 1. approved SDS or DDS & Test Plan
   - 2. updated RTM
How design level we will develop?

- PM, BA, TL and CUS will have a meeting to choose SDS or DDS.
- We can use SDS for High Level Design.
- We can use DDS for Detail Design.
Develop High Level Design - SDS

- TL defines for following sections and documents into Software Design Specification (SDS)
  - Design Considerations
  - System Level Desired Behaviour
  - Logical Representation of the Architecture
  - Architectural Component Overview
  - Process Architecture (Optional)
  - Deployment Architecture (Optional)
  - Open Issues

- TL updates RTM

- TL reviews High Level Design, RTM with SE, BA

Refer to 4+1view-architecture guideline for more info
Develop Detail Design - DDS

- SE and TL defines for following sections to develops detail design for each identified component and documents into DDS:
  - Object Model
  - Scenarios
  - Class Summary
  - Algorithms
  - Interfaces
  - Error Handling
  - GUI Mockups

- TL updates RTM

- SE reviews the final DDS, RTM with TL, BA and get approval from PM by formal or informal meeting.
Define Test Case and Test Plan

- SE defines Unit Test section in Test Plan (optional)
- QC develops Test Plan & Test Cases and send to TL and/or PM for review.
- PM and TL review and approve the Test Plan.
In this section, we have learnt about:

- **High Level Design activity**: to describe the overall application in various view (System, Logical, Architectural, Process view) and scenarios

- **Detail Design activity**: To develop the design solution for the application in sufficient detail Object model, Algorithms, Scenarios, Interfaces, Dependencies, Error Handling, GUI Mockup)

- **Develop Test Plan & Test Cases activity**: to prepare Test Plan for Unit & Integration Test and to develop Integration Test Cases
Implementation
Implementation

The purpose of software implementation is to produce executable software units and combine them to an integrated software items that properly reflect the software design.

Involved Role

- Project Manager
- Business Analyst
- Technical Lead
- Software Engineer
Implementation Workflow

1. SDS
2. Project Schedule
3. Coding Standard
4. Team Detail Plan
5. Build Plan (optional)

1. Coding Standard
2. Team Detail Plan
3. Build Plan (optional)
4. Code Review checklist

1. Source Code
2. Updated RTM
3. Unit Test Report
4. Code Review Log

1. Updated Project Schedule
2. Test Plan
3. Integration Test Report
4. Build Plan

1. Integration Test Report
2. Final Build
Prepare for Implementation

- PM creates Project Schedule
- TL selects coding standards
- TL decomposes the SDS into work items and put it to Team Detail Plan depend on Project Schedule.
- TL creates Build Plan (optional)
Coding & Unit Test

- SE executes works item in Team Detail Plan
- SE develops & runs Unit Test Cases and report the result in Unit Test Report
- SE reviews and analyzes unit test result against unit test criteria in Test Plan
- SE conducts code review with team using code review checklist
- SE fixes defect by opening new works item in Team Detail Plan.
- SE updates RTM to ensure its conformance to the design
Integrate & Integration Test

- SE establishes build environment and integrate components
- SE runs Integration Test Case and reports the results in Integration Test Report
- TL reviews and analyzes the Integration test results against Test Criteria in Test Plan
- SE fixes defect by opening new WI in Project Schedule.
- SE updates RTM to ensure its conformance to the design
- SE sends the final build to Testing team.
In this section, we have learnt about:

- **Prepare for Implementation activity**: to select coding standard & decompose SDS into work items

- **Coding & Unit Test activity**: to perform coding and conduct unit test by developing & running unit test cases then reporting the result

- **Integrate & Integration Test activity**: to integrate components, conduct integration test by running integration test cases and reporting
Testing
Testing

- Testing is an activity performed for evaluating product quality, and for improving it, by identifying defects and problems.

- Testing is performed to verify that the completed software package functions according to the established requirements

- Project Manager, Technical Lead and Software Engineer are 3 key roles which participate in this process
Testing Workflow

1. Determine Test Strategy
2. Update Test Plan
3. Develop TestCases
4. Execute Testing & Report

Technical Lead

Technical Lead

Software Engineer

Software Engineer

Technical Lead

Begin

1. SRS
2. Test Strategy
1. Test Plan
1. Test Plan
1. Feature Breakdown table
2. System Test Cases
3. Updated RTM
1. System Test Cases
1. System Test Suites
2. Defects
3. Test Reports

End

TMA Training Center (www.ttc.edu.com)

Version 1.0
Determine Test Strategy

- TL analyzes SRS and specifies features which will be tested and their test type using Test Strategy template
- TL defines Test Approach include following:
  - Manual or automation test
  - Test Type & its coverage (how many % for each test type)
  - Test cycle & duration
- TL determines test purpose and criteria for each type of test and document into Test Strategy which will be sent PM for approval.
Update Test Plan

- TL develops the Test Plan with referring to Test Strategy, Requirements for Test & Types of Test
- TL sends Test Plan for review & get approval from PM
Develop System Test Cases

- SE develops Feature Breakdown Table and send it to TL for review
- SE develops System Test Cases, update to Test Case document and send it to TL for review
- SE updates RTM
Execute Testing & Report

- SE selects System Test Cases to create System Test Suite then send it to TL for review
- SE sets up Test Environment then execute System Test Suite
- SE screens the defect and enter defect in Defect tracking tool if any during test execution
- SE updates Test Report for executed System Test Cases and send to TL
In this section, we have learnt about:

- **Determine Test Strategy activity**: is to define Test Approach, Test Purpose and Test Criteria for each kind of test
- **Develop Test Plan activity**: is to plan for system testing activities
- **Develop System Test Cases activity**: is to generate Test Cases
- **Execute Testing & Report activity**: is to implement System Testing and report Test result
Summary

- General activities of software development are: Requirement, Design, Implementation, Testing, Deployment & Maintenance
- Requirements engineering is the process of developing a software specification.
- Design and implementation processes transform the specification to an executable program.
- Testing involves checking that the system meets to its specification and user needs.
## Document Templates

<table>
<thead>
<tr>
<th>Document</th>
<th>Template</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requirement Repository</td>
<td>Requirement Repository template</td>
</tr>
<tr>
<td>SRS</td>
<td>Software Requirement Specification template</td>
</tr>
<tr>
<td>RTM</td>
<td>RTM template</td>
</tr>
<tr>
<td>Test Strategy</td>
<td>Test Strategy template</td>
</tr>
<tr>
<td>Test Plan</td>
<td>Test Plan template</td>
</tr>
<tr>
<td>Feature Breakdown table</td>
<td>Feature Breakdown table template</td>
</tr>
<tr>
<td>Test Case, Test Suite</td>
<td>Test Case template</td>
</tr>
<tr>
<td>Test Report</td>
<td>Test Report template</td>
</tr>
<tr>
<td>SDS</td>
<td>Software Design Specification template</td>
</tr>
<tr>
<td>C/C++ checklist</td>
<td>C/C++ checklist template</td>
</tr>
<tr>
<td>Java checklist</td>
<td>Java checklist template</td>
</tr>
</tbody>
</table>
Reference

- Requirement guideline
- Elicitation Techniques: http://www.swebok.org/ch2.html, section 3.2
- 4 + 1 View architecture guideline
- C/C++ Coding Guideline
- Java Coding Guideline
And...

Quality is measure of our dedication
Your questions, please!