



## **LABORATORY MANUAL**

**CZ3002: Advanced Software Engineering**

**2020~2021 Semester 1**

**Version 2.1.4**

**SCHOOL OF COMPUTER SCIENCE AND  
ENGINEERING  
NANYANG TECHNOLOGICAL  
UNIVERSITY**



**NANYANG**  
**TECHNOLOGICAL**  
**UNIVERSITY**

<b>DOCUMENT HISTORY .....</b>	<b>4</b>
Version 2.1.4 .....	4
Version 2.1.3 .....	4
Version 2.1.2 .....	4
Version 2.1.1 .....	4
Version 2.1 .....	4
Version 2.0.1 .....	4
Version 1.1 .....	4
<b>STATEMENT OF SCOPE OVERVIEW .....</b>	<b>5</b>
Scope .....	5
Outcome .....	5
Soft Skills .....	5
Deliverables .....	5
<b>DESCRIPTIONS OF FIVE LABS .....</b>	<b>6</b>
<b>Lab 1: Project Proposal and Project Requirements .....</b>	<b>6</b>
1. OBJECTIVE .....	6
2. LABORATORY .....	6
3. EQUIPMENT .....	6
4. INTRODUCTION and SCOPE .....	6
5. EXPERIMENT .....	8
6. DELIVERABLES .....	9
7. REFERENCES .....	9
<b>Lab 2: Software Quality Management .....</b>	<b>10</b>
1. OBJECTIVE .....	10
2. LABORATORY .....	10

3.	EQUIPMENT .....	10
4.	INTRODUCTION and SCOPE.....	10
5.	EXPERIMENT.....	10
6.	DELIVERABLES .....	11
<b>Lab 3: Project Planning and Prototype Construction.....</b>		<b>12</b>
1.	OBJECTIVE .....	12
2.	LABORATORY.....	12
3.	EQUIPMENT .....	12
4.	INTRODUCTION and SCOPE.....	12
5.	EXPERIMENT.....	12
6.	DELIVERABLES .....	13
<b>Lab 4: Configuration Management.....</b>		<b>14</b>
1.	OBJECTIVE .....	14
2.	LABORATORY.....	14
3.	EQUIPMENT .....	14
4.	INTRODUCTION and SCOPE.....	14
5.	EXPERIMENT.....	14
6.	DELIVERABLES .....	15
<b>Lab 5: Software Testing and Capability Maturity .....</b>		<b>16</b>
1.	OBJECTIVE .....	16
2.	LABORATORY.....	16
3.	EQUIPMENT .....	16
4.	INTRODUCTION and SCOPE.....	16
5.	EXPERIMENT.....	17
6.	DELIVERABLES .....	17
<b>APPENDIX.....</b>		<b>19</b>

## Document History

### **Version 2.1.4**

Creator: Althea Liang  
Created on: Aug.1, 2020  
Discussion: Althea Liang, Lane Shen

### **Version 2.1.3**

Creator: Althea Liang  
Created on: Aug. 16, 2019  
Discussion: Althea Liang, Lane Shen, Yi Li

### **Version 2.1.2**

Creator: Althea Liang  
Created on: Aug. 06, 2019  
Discussion: Althea Liang, Lane Shen

### **Version 2.1.1**

Creator: Althea Liang  
Created on: Jan.10, 2018  
Discussion: Althea Liang, Lane Shen

### **Version 2.1**

Creator: Althea Liang  
Created on: Aug. 9, 2017  
Discussion: Althea Liang, Lane Shen

### **Version 2.0.1**

Creator: Althea Liang  
Created on: Jan. 5th, 2017  
Discussion: Althea Liang, Lane Shen

### **Version 1.1**

Creator: Lane Shen  
Created on: Aug. 16th , 2016  
Discussion: Lane Shen, Althea Liang

# Statement of Scope Overview

In this page, we give an overview of the scope of the labs of module CZ3002. Every individual student must take a few minutes to read this one-page statement carefully in order to have an overall understanding of the scope and the deliverables of the course labs of CZ3002.

## Scope

Course project is an important part of the coursework in this course and your project mark takes up **50%** of your total mark of the course. The scope of the labs is an integral part of the scope of the course. It, in particular, helps you to achieve excellence in such graduate attributes as Design/development of Solutions and Modern Tool Usage.

Throughout the 5 labs of the course, the students are expected to first propose a software development project and then to go through the entire planning phase and partially the execution phase of their proposed project. The tasks of the labs consist of generating all kinds of the required software engineering assets based on learning the topics in lectures, studying and adapting the templates provided in NTULearn, discussing with the team the details of your project and of your project team, and coding and performing other practical work in a solid way.

## Outcome

After the 5 labs, the students will be capable of applying advanced *software engineering AND management techniques and practices* to the proposed large-scale software projects. Accordingly, students are to use appropriate tools for the development of a product prototype and the management of the project. Both parts are considered inseparable and important.

## Soft Skills

Throughout the 5 labs, soft skills are expected to be well exercised by all. You will be doing a team project in the context of labs of a course. Efficient team collaboration, successful teamwork and good self-motivation and management within the team shall start from the moment when a team is being formed to the moment the course ends and shall be at a highest priority.

## Deliverables

Deliverables of the labs include

1. Documents in all 5 labs
2. Demonstration and delivery of a live prototype (demo to be held at in-class session of lab 4)
3. Project presentation to be held at in-class session of lab 5
4. Backlogs and meeting minutes (One backlog file per lab and at least two meeting minutes per lab)
5. Peer review report

## Descriptions of Five Labs

### ***Lab 1: Project Proposal and Project Requirements***

#### **1. OBJECTIVE**

- 1) Form your team and establish your group members' roles and responsibilities.
- 2) Set up a MediaWiki site for group project, it will be used for collaborative work, e.g. sharing, updating knowledge.
- 3) Decide on what to build and write up the project proposal.

#### **2. LABORATORY**

This lab will be conducted in the Software Lab 3 (N4-B1C-14) in SCSE.

#### **3. EQUIPMENT**

PCs (Windows 7/8)

- MediaWiki
- Version Control System, e.g., SVN

#### **4. INTRODUCTION and SCOPE**

In this first lab, the scope is for you to build your team and propose your own software development project for the module. Below, we give an introduction on the supporting software that you will have to make use of in all the follow labs. Get familiar with these supporting software in class. Your in-class discussions on team building and project to be proposed will be recorded in meeting minutes. You will continue to fix more details of your project proposal after class. Pay attention to the items that are to be delivered before the next lab in section 6.

##### ***4.1 Tools: Wiki***

A **wiki** is software that allows users to create, edit, and [link](#) web pages easily. Wikis are often used to create [collaborative websites](#) and to power community websites. They are being installed by businesses to provide affordable and effective [Intranets](#) and for [Knowledge Management](#). [Ward Cunningham](#), developer of the first wiki, [WikiWikiWeb](#), originally described it as "the simplest online database that could possibly work". One of the best known wikis is [Wikipedia](#).

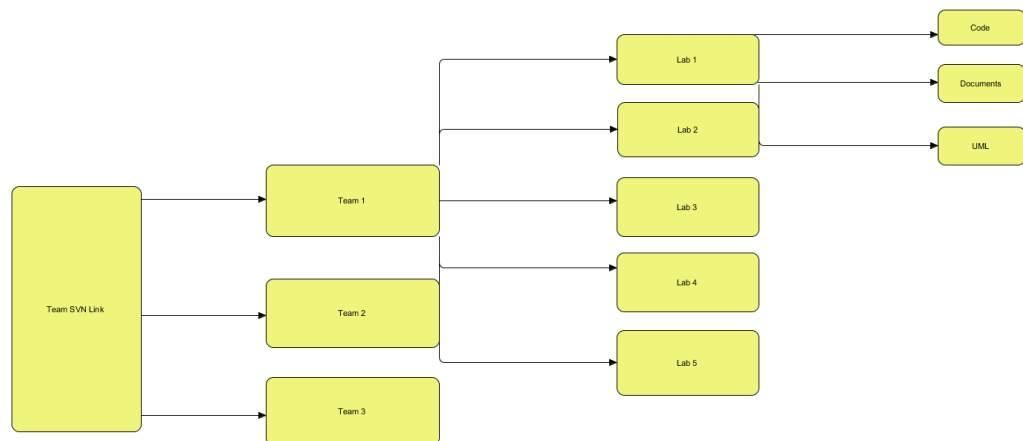
MediaWiki is [free](#) server-based software which is licensed under the [GNU General Public License](#) (GPL). It's designed to be run on a large server farm for a website that gets millions of hits per day. MediaWiki is

extremely powerful, scalable software and a feature-rich wiki implementation, that uses **PHP** to process and display data stored in its **MySQL** database.

When a user submits an edit to a page, MediaWiki writes it to the database, but without deleting the previous versions of the page, thus allowing easy reverts in case of vandalism or spamming. MediaWiki can manage image and multimedia files, too, which are stored in the file system. For large wikis with lots of users, MediaWiki supports caching and can be easily coupled with Squid proxy server software.

#### 4.2 Tools: SVN

SVN is a software versioning and revision control system distributed as free software under the Apache License. Developers use Subversion to maintain current and historical versions of files such as source code, web pages, and documentation. The structure of your SVN has been pre-created as shown below. You should save your files accordingly.



#### 4.3 Backlog

A backlog is a list of features or technical tasks which the team maintains and which, at a given moment, are known to be necessary and sufficient to complete a project or a release:

- if an item on the backlog does not contribute to the project's goal, it should be removed;
- on the other hand, if at any time a task or feature becomes known that is considered necessary to the project, it should be added to the backlog.

These "necessary and sufficient" properties are assessed relative to the team's state of knowledge at a particular moment; the backlog is expected to change throughout the project's duration as the team gains knowledge. The backlog is the primary point of entry for knowledge about requirements, and the single authoritative source defining the work to be done.

#### 4.4 Meeting Minutes

Meeting minutes are the instant written record of a meeting. They typically describe the events of the meeting and may include a list of attendees, a statement of the issues considered by the participants, and related responses or decisions for the issues. In details, the meeting minutes should at least include the following items:

- Date, time, venue and duration
- Attendees and non-attendees
- Chair of the meeting being identified
- Check and approve the last meeting minutes
- Topics/issues being discussed
- Solutions/actions being decided
- Who should take what actions
- When should each action be finished
- The next meeting being planed
- The minutes should be agreed by all participants and signed by the chair
- The minutes will be reviewed/checked and approved in the next meeting

## 5. EXPERIMENT

### 5.1 Setup Project Team

1. Form a project team with 6-7 students per team. Lab supervisor may adjust the members of the teams based on the specific situation of the lab group.
2. Assign project team roles. Each member may take one or two roles in the list below.
  - 1) Project Manager (Overall delivery of product),
  - 2) QA Manager (Overall product and process quality, implementation of QA processes),
  - 3) QA Engineer (Devise test plans, conduct tests),
  - 4) Lead Developer (Overall technical lead, responsible for technical aspects of product release),
  - 5) Front-end Developer (Android front-end programming. Participate in the entire SDLC, generating work products including documentation, source code, unit and integration tests)
  - 6) Back-end Developer (Server, application and database programming. Participate in the entire SDLC, generating work products including documentation, source code, unit and integration tests)
  - 7) Release Engineer/Manager (Create baselines and build and integrate changes for delivery. Manage releases of the product prototype.)
3. Register your team online so that the lab technicians will setup lab environment including access right to Wiki and SVN accordingly.
4. Please finish the first 3 steps in the first half of the session so that you have time to discuss the project and the project proposal in the team.

### 5.2 Hold the team meeting and complete the meeting minutes

In the meeting, the deliverables of this lab are discussed. Tasks and deadlines are identified and the tasks should be assigned. By end of the lab session, the minutes should be shown to the lab supervisor and submitted to the Wiki (for the first lab, if the Wiki access is not ready yet, the minutes should be submitted after the access is assigned but the minutes must be assessed by the lab supervisor). In the minutes, the preliminary idea of the system should be addressed and the system architecture should be described.

### 5.3 Setup Development Platform

1. Identify development platform and runtime platform (if different from the development platform) based on the project proposal
2. Check availability of the required software
3. Install the required software

### 5.4 Update the wiki page

Different from traditional web pages, only web host is able to modify the webpage contents. Wiki provides a flexible way that all registered users can modify the web contents in a direct way.

By clicking the “Edit” tab at the top of page, you are able to modify the whole page, or by clicking the “Edit” in every paragraph, you are able to modify a single paragraph.

For all the editing issues, please check the help content at [<http://www.mediawiki.org/wiki/Help:Contents>].

Please find the answers for the following questions:

1. how to make headings with different levels
2. how to make italic and bold fonts
3. how to add an image

### 5.5 Write Project Proposal

Project proposal is a document that is submitted to a business customer or an organization for acceptance and approval. The proposal describes the problem to be solved and explains the resulting benefits to the customer or the public.



A use case model (including use case diagram and accompanying use case description) can help you to identify the features and functionalities included in your proposed system.

In addition, innovative ideas that make your system different from existing similar systems, data collection mechanisms and data analysis using the collected data are encouraged to be included in your proposed system.

### ***5.6 Update the backlog excel file***

The backlog excel file of the team should be used and updated according to the task assignment and progress of each lab.

## **6. DELIVERABLES**

The following items should be delivered by the end of the in-class lab session:

1. Team registration at the lab computer system
2. Meeting minutes of the in-class discussion

Have the following ready (hosted/uploaded/compiled) by the next lab session:

3. Wiki: Team Information (names, roles, emails, phone numbers)
4. Wiki: Project proposal
5. Wiki: Use case model
6. Wiki: Backlog
7. Wiki: Meeting minutes

## **7. REFERENCES**

[1] MediaWiki: <http://www.mediawiki.org/wiki/MediaWiki>

[2] SVN: <http://svnbook.red-bean.com/>

[3] Backlog: <http://guide.agilealliance.org/guide/backlog.html>

## **Lab 2: Software Quality Management**

### **1. OBJECTIVE**

- 1) Setup quality assurance for the project development.
- 2) Formalize quality processes.
- 3) Design the prototype system to confirm the requirements of the proposed system.

### **2. LABORATORY**

This lab will be conducted in the Software Lab 3 (N4-B1C-14) in SCSE.

### **3. EQUIPMENT**

- PCs (Windows 7/8)
- Version Control System, e.g., SVN

### **4. INTRODUCTION and SCOPE**

In the previous lab, your team has proposed your software development project and should have obtained your lab supervisor's approval on the project. In this lab, the scope is for you to hold in-class *discussions* on the key points concerning planning for the quality of the software product you have proposed and document the key points in the meeting minutes. You should also start discussing what prototype you are building for the product including the main features, basic performance considerations and the tools to be used to support development. After the lab, you shall do further study and finalize your quality management strategy in your Quality Plan. You may choose to start building the actual prototype at this time.

You should emphasize a quality development *process*, in the context of your development project.

As required by the Verification Process Area, peer reviews (review meetings) are to be planned. The QA team will follow up on the required action and ensure that the issues raised are addressed and closed.

All of the processes and workflow for quality management should be formalized and described in the Quality Plan.

Prototype system can help to refine and validate the requirement on the entire product. High level design should be conducted in this lab for the preparing the project plan thereafter.

### **5. EXPERIMENT**

### ***5.1 Hold the team meeting and complete the meeting minutes***

In the meeting, the deliverables of this lab are discussed. Tasks and deadlines are identified and the tasks should be assigned. By end of the lab session, the minutes should be shown to the lab supervisor and submitted to the Wiki. In the minutes, the basic description of the software, hardware and other parts of the proposed system should be addressed and a list of points on quality management should be described. In addition, the plan for developing the prototype system should be briefly introduced.

### ***5.2 Complete System Requirement Analysis***

Given the requirements of the system, determine your goals, objectives and strategies:-

- What is the main purpose and direction of the application/system?
- What are the critical constraints of the application/system?
- What are the major features of the application/system?

Use the use case model to help you to capture high-level functional requirements and refine them further. A prototype system can help you to estimate the complexity and required technologies of the functionalities and features of your system. You can start your initial design of your prototype system based on the use case model before the implementation in the next lab.

System requirements are to be specified in such a way that they are verifiable / testable. Requirements must also be trace-able from their expression in a system requirement specification to their realization in software requirement specifications, design documentation, source code and test cases.

### ***5.3 Start Building a Prototype***

Build a prototype using a development framework (IDE). A prototype system should briefly demonstrate the main features of your system. In addition, it should be a working system including frontend and backend components, if any.

The development of the prototype shall begin with the most distinguishing feature/component of your product. The development work must be in a collaborative way by at least three developers.

Along the way, use the prototype to validate the correctness of the requirements, seeking clarification from the customer, if necessary. While building the prototype, the development team must also validate that the customer requirements can be met in the target environment.

The prototyping process can also be used to address technical risks, such as developers' insufficient understanding of the Development Framework, the tools and the best practices that you are about to adopt in the development, and the target environment constraints.

### ***5.4 Prepare Quality Management***

The quality models learned from lecture and quality management on different stage of the development lifecycle should be discussed in the team. The quality plan and practices on different stage of the development lifecycle should be documented.

The review meetings shall be arranged. Review results shall be documented and followed through.

## **6. DELIVERABLES**

The following items should be delivered by the end of the in-class lab session:

1. Meeting minutes of the in-class discussion

Have the following ready (hosted/uploaded/compiled) by the next lab session:

2. Wiki: System requirement specification
3. Wiki: Quality Plan
4. Wiki: Backlog
5. Wiki: Meeting minutes

## ***Lab 3: Project Planning and Prototype Construction***

### **1. OBJECTIVE**

- 1) Define processes for the project development life cycle.
- 2) Develop a prototype system to form the basis of the software development of the full product.
- 3) Prepare Project Plan.
- 4) Risk Management

### **2. LABORATORY**

This lab will be conducted in the Software Lab 3 (N4-B1C-14) in SCSE.

### **3. EQUIPMENT**

- PCs (Windows 7/8)
- IDE, Development Framework, Emulator

### **4. INTRODUCTION and SCOPE**

In the last lab, your team has planned for the quality management for your product. In this lab, the scope will be for you to continue to plan for other aspects of the project, e.g. including identifying key engineering activities, tasks and their deadlines. You should also continue discussing the prototype building among the developers. These should be documented in your meeting minutes. After class, you should continue to study further in order to complete scheduling of the entire proposed project. In parallel, you should have kicked off constructing the prototype and spend inclusively three weeks to one month in constructing the prototype to demonstrate the software product.

Project plan includes many activities such as effort estimation, duration of the development, lifecycle selection, resource assignment, and scheduling, etc.

An important activity during planning is to plan the risk management so that if any predicted risk really occurs, the corresponding actions would be taken according to the plan.

### **5. EXPERIMENT**

#### ***5.1 Hold the team meeting and complete the meeting minutes***

In the meeting, the deliverables of this lab are discussed. Tasks and deadlines are identified and the tasks should be assigned. By end of the lab session, the minutes should be shown to the lab supervisor and

submitted to the Wiki. In the minutes, the five elements for software estimation should be included and a list of risks to be addressed should be identified as well as the status of prototype system and any update should be reported.

### ***5.2 Continuing and Completing Prototyping***

Based on the discussions on prototyping in Lab 2, the develop team of the project team shall have gone well into prototyping of the software product in the proposed project.

The entire team shall discuss several tools for development team to use and several “best practices” well accepted by the industry for the development and entire team to adopt. Especially, the development team must start using required tools including SVN (or other version control software such as Git) during the development. Development team shall also consider start practicing “best practices” of continuous integration to enable future configuration management planning and requirements.

### ***5.3 Prepare Project Plan***

Planning methods such as the software estimation techniques, critical path analysis methods, bar chart scheduler should be used for preparing the project plan.

In particular, the Development Team needs to provide time estimations based on their experience during prototyping. With this input, the Project Manager can then perform resource allocation and plan the schedule so as to work within the various constraints and deliver the requirements specified within the delivery timelines.

Subsequent project monitoring and control involves reviewing the project progress on a regular basis and taking corrective action to bring the actual project implementation as close as possible to the planned.

Significant deviation which may delay delivery must be flagged and brought to the customer’s attention immediately.

### ***5.4 Prepare Risk Management***

Identify the possible risks during the project development, prepare the risk management strategies.

## **6. DELIVERABLES**

The following items should be delivered by the end of the in-class lab session:

1. Meeting minutes of the in-class discussion

Have the following ready (hosted/uploaded/compiled) by the next lab session:

2. Wiki: Project Plan
3. Wiki: Risk Management
4. Demo: Prototype ready for simple demo and support for the design and lab results
5. SVN: Prototype related items, such as code, documents, slides, video clips, etc.
6. Wiki: Backlog
7. Wiki: Meeting minutes

## ***Lab 4: Configuration Management***

### **1. OBJECTIVE**

- 1) Decide configuration management.
- 2) Setup change management
- 3) Prepare release plan
- 4) Decide design strategies towards maintainability.

### **2. LABORATORY**

This lab will be conducted in the Software Lab 3 (N4-B1C-14) in SCSE.

### **3. EQUIPMENT**

- PCs (Windows 7/8)
- Version Control System, e.g., SVN

### **4. INTRODUCTION and SCOPE**

In previous labs, your team has not only drafted out a complete plan for your project, but also built a prototype of your project. In this lab, the scope is for you to demo your prototype to the lab supervisor and for you to prepare for the release and delivery of the product. Releases and delivery of the product are part of configuration management. In the lab class, you will discuss the key points concerning the control of changes and management of your releases. Please document your discussions in meeting minutes and do after-class study on configuration management for your project.

After this lab, the configuration management plan and the release plan should be documented.

The system design should be reviewed towards maintainability. As the development team prepares for system design, the QA team will prepare change management policies and procedures.

### **5. EXPERIMENT**

***5.1 Hold the team meeting and complete the meeting minutes***

In the meeting, the deliverables of this lab are discussed. Tasks and deadlines are identified and the tasks should be assigned. By end of the lab session, the minutes should be shown to the lab supervisor and submitted to the Wiki. In the minutes, the strategy for designing the software system should be included and a list of releases to be planned should be identified as well as the idea of change management should be described.

### ***5.2 Design for Maintainability***

After considering maintenance of the development software, the design of the system should be discussed towards better maintainability. Some design strategies, such as design in small, test driven development, use of architectural design patterns, frameworks and Service Oriented Architecture (Web Services), would be considered and decided for the system design and implementation.

### ***5.3 Configuration Management by Tool(s)***

Access to SVN tool server will be assigned to teams in the project. A frequent usage of this tool provided by the lab for configuration management is recommended as the history of using it and items managed by this tool will be assessed after the lab session is finished.

Identify and confirm configuration items to be checked in. Discuss and decide policies for configuration management measurement and activities.

Based on your experience and findings in prototyping of using tools and best practices, prepare as part of the configuration management plan how the team plan to use various tools and best practices.

### ***5.4 Prepare Change Management***

Discuss and set baselines for the project. Discuss the roles and procedures for change management after the baseline. Decide the change request form and change process workflow and state tracing system.

### ***5.5 Prepare Release Plan***

Software development can be iterative, with multiple releases. For example,

- Release 0.1, for baseline
- Release 1.0, for the first release of the complete product
- Release 2.0, for the extended features
- Release 3.0, ...

Release 1.x is a refinement of Release 1.0, including bug fixes and Release 2.0 includes new features compared to Release 1.0.

## **6. DELIVERABLES**

The following items should be delivered by the end of the in-class lab session:

1. Meeting minutes of the in-class discussion
2. Prototype demonstration

Have the following ready (hosted/uploaded/compiled) by the next lab session:

3. Wiki: Design report on software maintainability
4. Wiki: Configuration Management Plan
5. Wiki: Change management plan
6. Wiki: Release plan
7. Wiki: Backlog
8. Wiki: Meeting minutes

## ***Lab 5: Software Testing and Capability Maturity***

### **1. OBJECTIVE**

- 1) Prepare Test Plan
- 2) Generate test cases
- 3) Formalize project management processes

### **2. LABORATORY**

This lab will be conducted in the Software Lab 3 (N4-B1C-14) in SCSE.

### **3. EQUIPMENT**

- PCs (Windows 7/8)
- IDE, Development Framework, Emulator
- Version Control System, e.g., SVN

### **4. INTRODUCTION and SCOPE**

In the last lab, your team has completed project planning by configuration management planning. In this lab, the scope is for you to further your quality management planning by creating a systematic mechanism on managing process quality. Not only this, you will also exercise quality control by creating and running test cases against your prototype. After that, we will also arrange your team to present your project to the class.

In this lab, the QA team will formulate the test cases based on the requirements specified for the project. Test then code and test automation should be practiced in this lab.

By now you have experienced the following processes in earlier labs, you shall aim to prepare ourselves to fulfill CMMI Level 2 requirements in the four Process Areas:

- Requirements Management
- Project Planning
- Software Quality Assurance
- Configuration Management

In this lab, the team should come up with the process definitions based on the CMMI model.

It's time to show your quality process and quality product in the presentation! Let us have a fruitful ending of the course labs.



## 5. EXPERIMENT

### *5.1 Hold the team meeting and complete the meeting minutes*

In the meeting, the deliverables of this lab are discussed. Tasks and deadlines are identified and the tasks should be assigned. Because the lab session will hold the final presentation, this activity should be done as soon as possible so that you can start other activities. The meeting minutes should be submitted to the Wiki not later than the midnight of the Friday of the same week of this lab session. In the minutes, the functions to be tested before the final delivery and whether alpha and beta testing are performed for user acceptance need to be addressed.

### *5.2 Prepare Test Cases*

Design test cases using Black Box techniques, specifying the test steps

- Input parameters
- Operations to perform
- Expected output (result).

### *5.3 Prepare Test Plan*

Test plan including different tests (functional test, non-functional test, system test, user acceptance test, etc.) should be formalized and documented.

### *5.4 Define CMMI level 2*

So far, teams have gone through the activities of requirement analysis, quality assurance, project planning and configuration management, etc. from previous labs. Based on the CMMI model learned from lecture, tutorial, the four process areas listed above which are required in CMMI level 2 should be defined. This is the necessary step towards higher level of CMMI model.

### *5.5 Peer Review (done individually)*

Individual members should fill up the peer review form online to evaluate the contributions to the project and attendance for the project meetings and discussions of other team members. The content of the form is confidential so don't discuss it in the team.

## 6. DELIVERABLES

The following items should be delivered by the end of the in-class lab session:

1. Presentation: Project introduction and summary of work of each lab, achievements, issues, solutions, etc.
2. Meeting minutes of the "in-class" discussion (submitted by end of Friday of the same week)

Have the following ready (uploaded/mailed/compiled) for final grading by the Lab instructor:

3. Wiki: Presentation slides
4. Wiki: Test Plan
5. Wiki: Test Cases and Requirements Test Coverage report
6. Wiki: CMMI level 2 definition
7. SVN (Version Control System): All documentation checked in
8. Wiki: Backlog
9. Wiki: Meeting minutes
10. NTULearn: Submission of the peer review before the deadline

## 7. **REFERENCES**

- [1] CMMI for Development Version 1.2 Chapter 4 – Four Categories of CMMI Process Areas.
- [2] Process Area (CMMI): [http://en.wikipedia.org/wiki/Process\\_area\\_\(CMMI\)](http://en.wikipedia.org/wiki/Process_area_(CMMI))

## Appendix

# CZ3002 Pre-defined Optional Lab Projects (S1 AY 2020-2021)

### 1. A face-recognition-based attendance taking system

Taking attendance is needed in many teaching/learning activities. The current attendance taking system is not only requiring manual efforts but also vulnerable to low performance. We hope to have a new attendance taking system using AI-based face recognition/identification techniques for our courses for better results. To build a useful system, the teams are to take into practical factors of the actual application of the system into consideration; and explore and select suitable techniques to achieve expected better performances.

### 2. A platform to facilitate the learning-based social connectedness

A challenge for many students in learning is lack of necessary social connectedness with the peers. To maintain the connection with peers in the same course, a useful platform is expected. Particularly, this platform may keep the students being aware of the contents being taught, the responses of other peers in the same learning content, and opportunities to exchange on opinions or questions on the content. This platform is needed help them find an outlet to contribute to the learning society, to promote the student's interactions in a community and to improve their learning effectiveness.

### 3. A drawing game that train brains

Drawing is an active way of learning in the sense it trains our brain, but not everyone likes it and not everyone is good at it. However, innovative technology, such as the app "Drawing Something", turns drawing easy, fun and sociable. You may develop a new drawing game that can train the capabilities of the brains of elderly, e.g. retrospective/autobiographic memory, and at the same time keep them mentally or/and socially active.

### 4. Online Trail Making Test (TMT)

TMT is an online test game for cognitive training. TMT is a neuropsychological test which was originally used for assessing general intelligence. The subject is instructed to connect a set of 25 dots as quickly as possible while simultaneously maintaining accuracy. The TMT performance metrics (e.g. completion time and error rate) have clinical utility in diagnostic classification of cognitively healthy older adults and patients with dementia.

