

The Student's Guide of Capstone Project Document for the FPT University Students

(For IS ES, BLC, ATM Specializations Purposes)

--- * ---

I. General Introduction

1. Capstone Project's description

Develop of a significantly complex software system, employing knowledge gained from courses throughout the program including defining requirements, design, implementation, and quality assurance. Students must follow a suitable process model, pay attention to quality issues, and manage the project themselves, following all appropriate project management techniques. Success of the project has to deliver the stages on times.

Sample deliverables:

Students should deliver one or several iterations of a software system, along with all artifacts appropriate to the process model they are using. These would likely include a project plan (perhaps updated regularly and containing cost estimations, risk analysis, division of the work into tasks, etc.), requirements (including use cases), architectural and design documents, test plans, source code, and an installable software package.

Additional teaching considerations:

- It is anticipated that this course will not have formal lectures, although students are expected to attend progress presentations by other groups.
- It is suggested that students be required to have a 'customer' for whom they are developing their software. This could be a company, a professor, or several people selected as representing people in the potential market. The objective of the project would be to solve the customer's problem, and the customer would therefore assist the instructor in evaluating the work.
- It is strongly suggested that students work in groups of at least three, and preferably four or five, on their capstone project. Strategies must be developed to handle situations where the contribution of team members is unequal.

2. The IT Capstone Project Products

The Capstone Project Product is composed of 7 parts:

- Part 1: Project Introduction
- Part 2: Project Management Plan (PMP)
- Part 3: Software Requirements Specifications (SRS)
- Part 4: Software Design Description (SDD)
- Part 5: Software Test Documentation (STD)
- Part 6: Software User Guides (SUG)
- Part 7: Software Product (SP)

The content of each report is described as in the part III ([Project Document Templates](#))

II. Capstone Project Regulation

FPT University students must follow the regulations presented here during the time doing the Capstone project. (In this document CP stands for Capstone Project).

1. Common regulations

- All students of FPT University have to do a CP in order to get a bachelor certificate.

- Capstone project will be conducted in class form, carried out in group. Each class will be supervised by a teacher and divided into groups of 4-5 students.
- The CP is worth 10 credits.
- The time allotted for doing and presenting the final CP is the entire semester.
- All CPs have to be presented to a review committee assembled by the chancellor of FPT University advisor.
- Students must be present at all meetings with their supervisor according to a previously arranged schedule. They must also attend other meetings conducted by the group and supervisor.
- English is official language for all documents and for the CP presentation.

2. Condition to apply for CP

- Successfully complete below topics:
 - On-the-job training (OJT).
 - Project Management – PMG201
 - Software Architecture and Design – SWD391
- Must successfully complete at least 3 elective subjects of ES or IS

3. Reports of CP

The Capstone Project Document that is composed of 7 parts called Reports:

- Report No.1: Project Introduction
- Report No.2: Software Project Management Plan (SPMP)
- Report No.3: Software Requirements Specifications (SRS)
- Report No.4: Software Design Description (SDD)
- Report No.5: Software Test Documentation (STD)
- Report No.6: Software User Guides (SUG)
- Report No.7: Software Product (SP)

4. Milestone table of CP implementation

#	Deliverable	Notes
1	Project Introduction Document	<u>Due</u> : end of week 1 (Initiation) <u>Output</u> : Report 1
2	Project Plan	<u>Due</u> : end of week 2 (Plan & Requirement) <u>Output</u> : Report 2
3	Overall Requirement Description	<u>Due</u> : end of week 3 (Plan & Requirement) <u>Outputs</u> : (1) Report 3, in which the team provides full information for the part I & below contents for the part II (others to be completed later) - II.1 Product Overview - II.2 User Requirements - II.3.1 System Functional Overview (2) Updated Report 1, 2 (3) Project Schedule/Tracking
4	Overall Software Design Description	<u>Due</u> : end of week 5 (Software Design) <u>Outputs</u> : (1) Report 4, in which the team provides System & Database Design (2) Report 5, in which the team provides all the information, excepts for the Test Cases & Test Reports (parts II.4 & II.5) to be completed later

		<p>(3) Demo code package, include workable codes (following the system design the team has defined) & database script. For example: with a Web system, the students might choose some common functions as below to build the “code frame” to prepare for their implementation</p> <ul style="list-style-type: none"> - User Login: coming to the Home Page after login, coming back here if logging out from the header - User Register: linked from the login page - Home Page: blank content, full header, footer - User Profile: linked from the header (user’s context menu) - User Management (list, add, update, delete): with full header, footer (and maybe menu tree in sider) <p>(3) Updated Reports 1, 2, 3 (4) Updated Project Schedule/Tracking</p>
5	Software Package 1	<p><u>Due:</u> end of week 7 (Code Iteration 1) <u>Outputs:</u></p> <p>(1) Software Package 1</p> <ul style="list-style-type: none"> - Workable Source Codes & Database Script - Functional Requirements (in the Updated Report 3) - Class & Sequence Diagrams (in the Updated Report 4) - Test Cases & Reports: UT, IT, Defects List (in the Updated Report 5) <p>(2) Updated Reports 1, 2, 3, 4, 5 (3) Updated Project Schedule/Tracking</p>
6	Software Package 2	<p><u>Due:</u> end of week 9 (Code Iteration 2) <u>Outputs:</u> similar to the Software Package 1 above</p>
7	Software Package 3	<p><u>Due:</u> end of week 11 (Code Iteration 3) <u>Outputs:</u> similar to the Software Package 1 above</p>
8	Full Software Package	<p><u>Due:</u> end of week 13 (System Testing) <u>Outputs:</u></p> <p>(1) Full Software Package</p> <ul style="list-style-type: none"> - Workable Source Codes & Database Script - Functional Requirements (in the Updated Report 3) - Class & Sequence Diagrams (in the Updated Report 4) - Test Cases & Reports: ST, IT, Defects List (in the Updated Report 5) <p>(2) Updated Reports 1, 2, 3, 4, 5 (3) Updated Project Schedule/Tracking</p>
9	User Guides	<p><u>Due:</u> end of week 14 (Transition) <u>Outputs:</u></p> <p>(1) Report 6: User Guides (2) Updated Project Schedule/Tracking</p>
10	Final Package	<p><u>Due:</u> end of week 15 (Transition) <u>Outputs:</u></p> <p>(1) Report 7: Final Project Report (2) Final Project Products</p> <ul style="list-style-type: none"> - Database Scripts - Source Codes & Related Files - Test Documents & Defects List - Project Schedule/Tracking - ... <p>(3) Presentation File (Defense Slides)</p>

5. Evaluation of the Capstone Project

- The On-going Assessment (OGA) is collected from 7 reports with ratios as below. The detailed assessment for each report can be found in the Syllabus (the Assessment Matrix sheet). This assessment is given by the supervisor
 - Project Introduction (Report 1): 4%
 - Project Management Plan (Report 2): 8%
 - Software Requirement (Report 3): 16%
 - Software Design (Report 4): 18%
 - Testing (Report 5): 18%
 - User Guides (Report 6): 4%
 - Implementation (Report 7): 32%

The Thesis Defending Assessment (TDA) is given by each member of committee for the whole team and for each student after the final project presentation. Criteria for evaluating the presentation mark is as below (the details can be found in the Syllabus – the Assessment Matrix sheet). The TDA for each team member is the average of the marks given by each member of the committee.

- Project Introduction: 5%
- Project Management Plan: 5%
- Software Requirement: 15%
- Software Design: 10%
- Testing: 10%
- User Guides: 5%
- Implementation: 35%
- Presentation Skills: 5%
- Q&A: 10%
- Final Project Mark = $OGA * 50\% + TDA * 50\%$
- All marks are given on a scale of 0-10, precise to one place after the decimal. (For example: 6.7 marks)

III. Project Document Templates

1. Project Reports

1.1 Report 1 (Project Introduction)

This part is about the group of developers, the initial idea of the group's Capstone project, overview of similar existing solutions, and overview of existing methods, business outline, and limitations of the existing system. From that point, you will advance your knowledge/skill levels in the topic you have chosen, and benefits of expected system

Please see details in the template **Report1_Project Introduction**.

1.2 Report 2 (Project Management Plan)

This document mentions the detailed management plan for the project. Please see details in the template file **Report2_Project Management Plan** and the sample file **Report2_Sample Project Schedule** (the high-level schedule in this sample is following the RUP software development process which is the recommendation for your project, but not required)

1.3 Report 3 (System Requirement Specification)

In the document, both user requirements and software requirement are specified so that the project stakeholders would have the whole understanding about what the team would create in the project. Please find details in the template **Report3_System Requirement Specification**. Along with this report, the team also start preparing the Product Backlog & draft Iteration Backlogs following the template **Report3_Project Tracking** (details as mentioned in the milestone table above)

1.4 Report 4 (Software Design Document)

This document provides the details on the design for the software, including architecture design, detailed design and the database design. Please see details in the template ***Report4_Software Design Document***.

1.5 Report 5 (Software Test Documentation)

This documentation are the templates for testing related documentation. They include 3 templates with three different purposes as listed below;

- **Report5_Unit Test:** unit test case specification and tracking
- **Report5_Test Report:** integration, system, acceptance test case & tracking
- **Report5_Test Documentation:** testing plan and consolidation of the project testing information

1.6 Report 6 (Software User Guides)

This document describes the content of the final deliverable package, the installation guides and software user manual. Please see details in the template ***Report6_Software User Guides***.

1.7 Report 7 (Final Project Report)

This report includes the final updated project outputs as provided in the Part 1-6, provided following the template ***Report7_Final Project Report***

2. Templates for Tracking Project Status & Progress

2.1 Project Schedule & Tasks

To track the project schedule/time, the team need to prepare and track the project schedule (broke into main task level) using the Microsoft Project or Project Libre tool.

Sample schedule can be found in the provided sample schedule - ***Report2_Sample Project Schedule*** as mentioned above.

Detailed tasks can be tracked using any team-convenient tools (GitLab, Trello, Asana, etc.)

2.2 Tracking Project Information

The Project team should maintain a tracking database which includes below information:

- WBS: for tracking project works
- Issues: for tracking the project issues
- Defects: for tracking the product defects
- Q&A: to track the questions & answers during project progress

The details of each sheet can be found in the template file ***Report3_Project Tracking***.

The team can convert those files to Google sheet and share among the team and the supervisor so that all people can access, track information and have real-time information.

You can also track that information using any team-convenient tools (GitLab, Trello, Asana, etc.)