CSE3/5PRA & CSE3/5PRB

Industry Project - 2019

Handbook

Dr. Torab Torabi

Dr. Scott Mann

Dr. Amin B. Abkenar

Florian Puersch

2019Table of Contents

[Chapter 1 – Industry Project 5](#_Toc5883427)

[1.1 Introduction 5](#_Toc5883428)

[1.2 Audience Description 5](#_Toc5883429)

[1.3 Purpose 6](#_Toc5883430)

[1.4 Scope 6](#_Toc5883431)

[1.5 Organisation of this Handbook 7](#_Toc5883432)

[1.6 Acknowledgement 7](#_Toc5883433)

[Chapter 2 – Industry Project Subject Organisation 8](#_Toc5883434)

[2.1 Introduction 8](#_Toc5883435)

[2.2 Lecture and Workshop Timetable 8](#_Toc5883436)

[2.3 Meetings and Workshops 9](#_Toc5883437)

[2.4 Staff and Responsibilities 10](#_Toc5883438)

[Chapter 3 – Process and Development Guidelines 12](#_Toc5883439)

[3.1 Introduction 12](#_Toc5883440)

[3.2 Project Operational Guidelines 12](#_Toc5883441)

[3.3 Meetings 13](#_Toc5883442)

[3.4 Academic Misconduct 15](#_Toc5883443)

[3.5 Account Usage 15](#_Toc5883444)

[3.6 Plagiarism 15](#_Toc5883445)

[Chapter 4 – Software Document Standards 16](#_Toc5883446)

[4.1 Introduction 16](#_Toc5883447)

[4.2 Structural Guide for Documents 17](#_Toc5883448)

[4.3 System Maintenance Document 20](#_Toc5883449)

[4.4 Reflection Component for Masters Students 26](#_Toc5883450)

[4.5 User Documentation 28](#_Toc5883451)

[Chapter 5 – Assessment Guidelines for Subject A 30](#_Toc5883452)

[5.1 Introduction 30](#_Toc5883453)

[5.2 Assessment Components 30](#_Toc5883454)

[5.3 Major Submission Assessments for Project A 33](#_Toc5883455)

[5.4 Major Meetings and Presentations Assessments 35](#_Toc5883456)

[Chapter 6 – Assessment Guidelines for Subject B(TBA) 41](#_Toc5883457)

[6.1 Sprint Review/Retrospective Meeting Assessment – 30% 42](#_Toc5883458)

[6.2 Sprint Review/Retrospective Individual Assessment – 70% 44](#_Toc5883459)

[6.3 Reflection Assessment for Masters Students 45](#_Toc5883460)

[6.4 Demonstration Assessment 46](#_Toc5883461)

[6.5 Final Presentation Assessment 48](#_Toc5883462)

[6.6 Source Code Submission Assessment 50](#_Toc5883463)

[6.7 System Maintenance Document Submission Assessment 51](#_Toc5883464)

[6.8 User’s Guide Submission Assessment 53](#_Toc5883465)

[References 54](#_Toc5883466)

[Appendix A : Statement of Authorship 55](#_Toc5883467)

[Appendix B : Statement of Effort 56](#_Toc5883468)

[Appendix C : Document Change Record 57](#_Toc5883469)

[Appendix D : Guidelines and Procedures for Major Submissions 58](#_Toc5883470)

[Introduction 58](#_Toc5883471)

[Where to Submit 58](#_Toc5883472)

[Electronic Submission 58](#_Toc5883473)

[Appendix E : Directory Structure for Final Submission 59](#_Toc5883474)

[Suggested Submission Directory Structure: 59](#_Toc5883475)

## Chapter 1 – Industry Project

### 1.1 Introduction

The Industry project enables students to experience real-life involvement in the planning and development of a medium sized software project. It utilises a teamwork approach to project management and software engineering and integrates previously learned skills and knowledge in design, programming, database, social, and ethical aspects.

The Industry Project consists of two 15-credit units. The first unit (CSE3PRA/CSE5PRA) covers software engineering process, team organisation and communication, project scope management, project time management, system models and design, project risk and quality management, usability engineering, software testing, project plan and business requirement delivery and software design delivery. The second unit (CSE3PRB/CSE5PRB) follows on from CSE3PRA/CSE5PRA and covers project cost management, project human resources and communication management, software re-use, various issues and trends in software development industries, prototype delivery and final deliverable presentation.

Project teams must be finalised by the first week of semester 1 (CSE3PRA/CSE5PRA). The project is completed over two semesters; students must enrol in both units in order to complete the Industry Project. The Industry Project has been developed and continuously improved since 1992. A number of projects developed during previous years have been implemented in industry, or awarded research grants, or won the first place in state-wide and nationwide university project competitions.

This handbook has been prepared to provide students with guidelines and processes that should be followed as an integral part of the Industry Project for management and development of a successful product. This handbook will evolve throughout 2019. Always use the latest version that is available through LMS.

### 1.2 Audience Description

Industry Project is offered to students majoring in computer science or double degree programs in the Department of Computer Science and Information Technology. As a major part of the project students have to manage, design, implement, test, and document a medium size industry oriented software project as part of a team.

This handbook is prepared for the students and should be used by all members of the project teams. It provides a guideline for the various activities for project management, software design, implementation, and software documentation. Students should follow the guidelines, standards, and assessment criteria in this handbook closely as part of the process they would follow during software development. This handbook is also used as a baseline for evaluation and assessment of teams and individual contribution of team members by project supervisors. The supervisors assess students and teams as how closely they follow the **processes** and **best practices** and **manage** the project development prescribed by this handbook and major software research organisations. Students are encouraged to research, develop, and follow processes and procedures for undertaking and managing their activities within their team.

### 1.3 Purpose

Industry Project aims to introduce the students to challenges that they would face in a software organisation. Students have to design, implement, document, and manage the development of a commercial size project in a team environment. Important elements of this exercise are:

1. Team communication and team management
2. Project management, risk analysis, configuration management, and scheduling of tasks and activities.
3. Analysis of the problem domain and elicitation of the business requirements (conceptual modelling).
4. Design of the system, which includes translating the business requirement into a software design that forms the blueprint for implementation. It is important at this stage to pay attention to user requirements, the flexibility and limitations of existing software libraries and tools, and the engineering approach for the design of reusable components, which are to be used for effective and efficient systems implementation and integration.
5. Implementation of product based on the design to produce a reusable and high quality software product.
6. Testing and software documentation of the system.

### 1.4 Scope

This handbook covers basic guidelines and standards, as well as internal assessments for the design and implementation of software systems in an educational environment. The handbook covers all aspects of software development from project management and software requirement to detail design and software documentation. The intention is to expose students to standards in analysis, design, and implementation of software systems and provide them with guidelines for software documentation. This handbook should be used along with the Scrum guide from scrum.org and the prescribed textbook.

It is beyond the scope of this handbook to cover the Software Engineering Standards in detail. Students are encouraged to refer to Standards in Software Engineering established by organisations such as IEEE, European Space Agency, Software Engineering Laboratory (SEL), Software Quality Institute (SQI), and Software Engineering Institute (SEI).

### 1.5 Organisation of this Handbook

The remaining chapters of this handbook will cover:

1. Project Organisation
2. Process and development guidelines
3. Software Document Standards
4. Assessment Guidelines

Chapter 2 contains information on project organisation. Chapter 3 provides a review of process and development guidelines. Chapter 4 provides templates and description of documents that have to be prepared by students. Chapter 5 provides the assessment criteria for all documentation, presentations, and final software product.

### 1.6 Acknowledgement

This handbook has been revised based on the previous handbook of Industry Project written by Dr. Torab Torabi. The authors would like to thank current and previous academic staff of Department of Computer Science and Information Technology including Professor Tharam Dillon, Professor Elizabeth Chang, Professor Wenny Rahayu, Associate Professor Jacob Cybulski, Associate Professor Karl Reed.

We also would like to thank the tutors and project supervisors who provided valuable comments during previous years. Finally, this version has been improved by numerous feedbacks from the students through the years.

## Chapter 2 – Industry Project Subject Organisation

### 2.1 Introduction

This section covers the syllabus, timetables, assessment criteria, and staff organization of the Industry Project Teaching Staff and their roles and responsibilities.

### 2.2 Lecture and Workshop Timetable



#### 2.2.1 Industry Project A

All lectures workshops are subject to change in sequence or may vary in content with latest material during the academic year.

#### 2.2.2 Industry Project B (To be updated)

|  |  |  |
| --- | --- | --- |
|  | Lecture / Seminar Topic | Workshop Topic |
| 1 | Subject Outline & Metrics  Team Dynamics | Interactive Workshop |
| 2 | Process Improvement  Interactive Seminar 1 | Usability Testing |
| 3 | Usability Engineering 1  Guest Speaker | JIRA Capture & Test Tracking |
| 4 | Usability Engineering 2  Guest Speaker | **No Workshop**  **(Demo Practice)** |
| 5 | **No Lecture / Seminar**  **(Demo)** | **No Workshop**  **(Demo)** |
| 6 | Software Quality  Interactive Seminar 2 | Documentation |
| 7 | Documentation  Guest Speaker | Automated Testing |
| 8 | Presentation Skills Seminar | Usability Execution |
| 9 | Project Evolution  Lessons Learned | Subject Retrospective |
| 10 | Closing Your Project  Consultation | Something Special |
| 11 | **No Lecture / Seminar**  **(Final Submission)** | **Presentation Consultation (Optional)**  **(Final Submission)** |
| 12 | **No Lecture / Seminar**  **(Final Presentation)** | **Final Presentation** |

### 2.3 Meetings and Workshops

Students will work on the project as part of a team and a supervisor will be allocated to the team. Every week, each team should attend at least one meeting to discuss the team’s progress and discuss difficulties and allocate new tasks. Every other week the team meeting will be supervised and assessed by the team’s supervisor. In these meetings, the team supervisor will check the team’s progress, provide feedback and assess the Individual’s and team’s progress. The team supervisor will assist the team with the issues related to the project management, project documentation, presentation and project-related issues. More information can be found in section 3.3 of this handbook.

There will be scheduled workshops throughout the year. These workshops are designed around the skills required for managing and developing the project. The schedule will be announced during the first lecture. The workshops will cover various aspects such as:

* Project Planning, Scheduling, and Management Tools
* Project Estimation and Risk Analysis
* Usability Engineering & Prototyping
* OO Design & Implementation
* Quality Management
* Development, Versioning and Testing Tools
* Configuration Management

It is mandatory to attend all team meetings and weekly workshops. Workshops activities will be marked during assessed sprint meetings. Failure to attend the sprint review meeting results in a mark of zero for that week’s internal assessment. Students who can’t attend the meetings for medical reasons must provide a certificate. If students have technical problems, they are encouraged to consult their tutor or the debuggers in the department.

### 2.4 Staff and Responsibilities

#### 2.4.1 Staff

##### Subject Coordinator

**Dr. Scott Mann**

|  |  |
| --- | --- |
| Office | BG237, Beth Gleeson, Bundoora |
| Email Address | S.Mann@latrobe.edu.au |

##### Lecturer

**Dr. Amin B. Abkenar**

|  |  |
| --- | --- |
| Office | NA |
| Consultation Time | Monday 10:00 am – 12:00 pm, BG Data Science Lab |
| Email Address | [a.abkenar@latrobe.edu.au](mailto:a.abkenar@latrobe.edu.au) |

##### Project Supervisors

|  |  |
| --- | --- |
| Sameer Muhammad | S.Muhammad@latrobe.edu.au |
| Alex Tomy | S.Tomy@latrobe.edu.au |
| Alistair Clough | [A.Clough@latrobe.edu.au](mailto:A.Clough@latrobe.edu.au) |
| Florian Puersch | f.puersch@latrobe.edu.au |
| Majed Alwateer | M.Alwateer@latrobe.edu.au |

#### 2.4.2 Consultations

Team supervisors provide consultation only for the teams allocated to them. The team supervisor will handle problems related to team dynamics. Clashes of opinions and work ethics are a reality in team-based projects. It is important to learn to deal with these issues in a mature way by providing evidence for one’s position. Problems within the team impacts on team productivity; failing to address these issues will result in low marks for weekly assessments. Any issues remaining unresolved after discussions in the team and with the team supervisor may be brought to the Industry Project Coordinator.

#### 2.4.3 Additional Help

All students are welcome to bring their questions and concerns to the team supervisors and lecturer. We are here to help you as much as we can. After Labs and consultations, if you still have questions, contact your team supervisor or lecturer.

## Chapter 3 – Process and Development Guidelines

### 3.1 Introduction

To provide the students the best practical experiences, the Industry Project designed to simulate the environment within a software development organisation. Industry Project staff, including support staff and resources within the department of Computer Science and Information Technology, play the role of the Software Development Company, the organisation responsible for commissioning the software development.

The Industry Project Lecturer plays the role of the company manager. They will present real project requirements from members of the community or university who will act as clients for the projects. The tutors who are also project team supervisors play the role of scrum masters. The project teams will act as subcontractors or development teams for this company.

### 3.2 Project Operational Guidelines

For a successful management and development of the project, it is highly recommended that students follow the following general guidelines:

#### 3.2.1 Overall Project Process

Students should use an Agile approach in developing Industry Projects. We recommend scrum, supported by Extreme Programming where appropriate. Teams should work on their allocated project in 2 week sprints, ideally implementing a small vertical slice of the solution in each sprint.

#### 3.2.2 Team Formation and Registration

Students should form a software development teams of 5 members, one of them will be appointed as the team contact. All teams must be formed and approved by the end of the first week of the first semester. Each team will be allocated to a supervisor. No change will be possible in the composition of teams during the project. Once you have formed a team you are required to register it. Team registration is done through a form on LMS. Fill it out and email it to your lecturer.

#### 3.2.3 Project Allocation

Normally, a few projects are selected from industry collaborators or clients. A project will be allocated to a team based on the team preference, skills and resources. While every effort is made to meet the team’s preferences the project allocation is decided by the lecturer and is not negotiable. Students will receive an overview of each project at the start of semester 1 which they will use to determine their preferences. The detailed project requirement will be given once the projects have been allocated to teams.

#### 3.2.4 Delivery of Presentations

There are a number of intermediate deliveries during the year-long Industry Project. At each stage, the project team should prepare a 20 to 30 minute presentation based on the current progress of the software development. All team members should prepare and give these presentations jointly. The assessment criteria for each of these presentations can be found in chapter 5.

#### 3.2.5 Joint Effort

Each team member should be able to learn and demonstrate all skills (Project Planning and Management, System Analysis, Software Design, Software Implementation/Coding, Software Testing, and Software Documentation, Presentation) involved in software engineering. Team members must continuously learn and improve so any new skill can be applied to all development activities in all stages of the project.

#### 3.2.6 Team Communication

Set up good communication links between team members. Meet regularly. Use collaboration tools such as Confluence, Jira, wikis, email, and phone to establish communication channels between team members. Good communication is key to the successful completion of the project and should be established early. Don’t challenge your problems alone. Learn with and from your team members. Also share your knowledge with each other.

#### 3.2.7 Last Minute Work

A major project submission cannot be completed in a last minute rush, it is simply too big and complex. Team members must work consistently and continuously. The team should review and revise the work during each week, to include in the next major submission.

#### 3.2.8 Document Submission

There are a number of major document submissions during the year-long project. Project members should carefully prepare the submissions based on standards given in the Software Document Standards (Chapter 4) and Assessment Criteria (Chapter 5) in this handbook. The submissions are a result of the individual contributions and collaboration of the team during the weeks preceding the submissions. To be able to improve the quality and the standard of the submission, selected team members can be chosen to review each major activity. For example, a chief tester will be in charge of implementing testing standards, coordinating and reviewing all activities and documentation for testing.

It is the students’ duty to take care of the following responsibilities:

* All project teams are required to submit their project documents/projects on the due dates through Confluence tool.
* Every project team is required to submit “Statement of Effort” along with every major submission.

#### 3.2.9 Don’t Waste Time

Review team's progress regularly and keep track of team members’ tasks. Spend time to teach each other different skills. Do not waste time arguing over minor details. Do not blame each other; instead concentrate on resolving the issues to complete the tasks.

### 3.3 Meetings

Teams are expected to hold reqular meetings per week. All teams members should attend the scheduled supervised/assessed meetings . The marking critera for assessments can be found in sections 5.5, 5.6 and 5.7. As you will be following an agile process, we have defined 3 different kinds of meetings that your team will hold in each 2 week sprint. The first kind is the Sprint planning meeting, the second is a general meeting and the final kind of meeting combines the sprint review with a retrospective. Teams are expected to hold 1 sprint planning meeting, 2 general meetings and 1 sprint review & retrospective meeting during their sprints. These meetings are explained below. Further information can be found in the scrum guide which can be found [here](https://www.scrum.org/Scrum-Guide) and on LMS.

All meetings must be documented in Confluence by taking (brief) meeting minutes.

#### 3.3.1 Sprint Planning Meeting

The first meeting in your sprint.

In this meeting your team looks to answer the following questions:

* What can be done in this sprint?
* How will we get the work done?

In this meeting, teams are expected to allocate an appropriate amount of items from their project backlog to their sprint backlog (or To-Do list). They have a look at what the items require them to do and how they will go about doing them.

Refer to pages 8 - 10 of the Scrum guide for more information.

#### 3.3.2 General Meetings

Meetings throughout your sprint.

These meetings will begin with a 15 minute stand-up meeting (sometimes called the daily scrum, page 10 of the Scrum guide), where team members answer the following three questions:

* What have I done since we last met?
* What will I do next?
* Is there anything stopping me from reaching the sprint goal?

Team members spend the rest of the meeting time discussing items on their agenda.The meeting will end with calculating the team temperature.

#### 3.3.3 Sprint Review & Retrospective Meetings

The last meeting in your sprint.

##### Sprint Review Section (Page 11, Scrum guide)

This meeting is held at the end of your sprint to update the product backlog and to determine which items in your sprint are “done”. Team members also use this meeting to discuss what went well, what problems they ran into and how they overcame them.

Team members demonstrate user stories and tasks that have been classified as done and answer general questions about their work. The team discusses what is up next, this will lead into the sprint planning meeting.

##### Sprint Retrospective Section (Pages 11 and 12, Scrum guide)

The team discusses how the last sprint went with regards to people, relationships, processes and tools. They highlight what went well and attempt to address any issues related to these topics.

### 3.4 Academic Misconduct

The University clearly explains student’s responsibilities and rights in undergraduate/postgraduate handbook and also on the university’s web page <http://www.latrobe.edu.au/students/>. Please refer to the above references regarding your right and responsibility. Not knowing these responsibilities is not an excuse.

### 3.5 Account Usage

Only departmental and university account and emails should be used during the project for official communication between the team and the staff. Rules regarding these resources MUST always be observed.

### 3.6 Plagiarism

Copyright and non-original material must be properly referenced. Failure to do so may result in charges related to Academic misconduct: <http://www.latrobe.edu.au/students/admin/academic-integrity>

## Chapter 4 – Software Document Standards

### 4.1 Introduction

In this section the standards for all the submissions of Industry Project is presented. The templates and the descriptions are only guidelines to provide a basis for the content of these submissions. The guidelines may change depending on technology, research and market expectation.

#### 4.1.1 What is a documentation standard?

A documentation standard is an agreed template in the software industry worldwide, against which the system documentation must be written during the course of system development.

The following templates will be provided within this documentation:

* User Document
* System Maintenance Document

#### 4.1.2 Who are these standards for?

This compendium of documentation standards has been developed for use by Industry Project students within the Department of Computer Science and Information Technology, La Trobe University. It is intended that students should follow these standards during their system documentation.

#### 4.1.3 How do I use them?

Documentation standards provide a template against which documentation associated with any system development can be designed. Obviously as the range of possible system developments is immense, it is not possible for all parts of any documentation standard to be applicable to all development projects. As a consequence, these standards are proposed as a target against which the documentation to be written can be negotiated with the supervisor/potential user of the system being built. If you are directed to write to one of these standards, but do not believe that certain sections are applicable to your particular development, it is your responsibility to discuss with the supervisor/potential user, for the non-inclusion of those sections.

#### 4.1.4 Where can I get further information?

A variety of texts and proceedings discuss techniques and notations that can be used in the course for preparing documentation. Some of these texts are as follows:

* Australian Standards for Software Quality Management System.
* IEEE Standards - for Software Engineering, ANSI/IEEE, NY (<http://standards.ieee.org/software/>)

The documentation standard presented here ONLY provides templates for the documentation that will be prepared by students. It is assumed that students will consult other references such as the above to become familiar with specific techniques and notations, which must be used in the course of preparing documents.

### 4.2 Structural Guide for Documents

This structural guide can be used for the most of the major documents, especially recommended for both the *User Document* and the *System Maintenance Document*.

A common misconception of the Agile process is that documentation is not required. Agile advises team members to create documents as much as they are required, instead of creating long, detailed documents.

|  |
| --- |
| Title Page  Front Section  Copyright Information  Warranties and Contractual Obligations  Table of Changes  Table of Contents  Table of Figures |
| 1.0 Introduction  Content Section  1.1 System Overview  1.2 Audience Description  1.3 Applicability Statement  1.4 Purpose Statement  1.5 Document Usage Description  1.6 Conventions  1.7 Change Log  2.0 – N.0  <*Remaining contents of the document >*  *<Every documentation submission will largely vary in this section>* |
| Appendices  Back Section  Bibliography  Glossary of Terms  Reference Materials  Index  Statement of effort  Inspect and Verified by (Quality Assurance) |

#### 4.2.1 Front Section

###### Title Page

On the Title Page you must include at least the following information in the format provided in the standard pages for the Industry Project (Refer to Appendix C).

###### Copyright Information

When restrictions apply to using or copying the document or software product, describe these on the title page or immediately following the title page.

###### Warranties and Contractual Obligations

Specify any warranties and contractual obligations or disclaimers.

###### Table of Contents

Include a table of contents. For single-volume documents, this should be a comprehensive table of contents for the whole document.

###### Table of Figures

Include a table of all figures in the document, sequentially numbered.

#### 4.2.2 Content Section

###### 1.0 Introduction

1.1 System Overview

This section should provide a brief description of the system’s operational components. A *block diagram* that shows the major components of the system and their interconnections can be included here.

1.2 Audience Description

Describe the intended audience (e.g.: experience level and previous training expected of the user) of the system. If different for different sections, indicate the intended audience for each section.

1.3 Applicability Statement

State the software version covered and the environment in which the software runs.

1.4 Purpose Statement

Summarise the purpose of the software. Include intended applications of the software.

1.5 Document Usage Description

Describe what each section of the document contains, its intended use, and the relationship between sections. Also provide any other directions necessary for using the document.

1.6 Conventions

Summarize symbols, stylistic conventions, and command syntax conventions used in the document.

1.7 Change log

Summarise any changes that are made in each revision of your document. An example of this can be found in [chapter 1.7](#_1.7_Change_log) of this handbook.

###### 2.0 – N.0 <Remaining contents of the document>

***<Every documentation submission will largely vary in this section>***

#### 4.2.3 Back Section

###### N+1 Appendices

Include any supporting material, arranged for ease of access (e.g: sample files, reports, etc).

###### N+2 Bibliography

List all publications specifically mentioned in the text.

###### N+3 Glossary of Terms

List alphabetically definitions of all terms, acronyms and abbreviations used.

###### N+4 Reference Materials

List all external documents and sources of information referenced by the document.

###### N+5 Index

Develop an alphabetical index based on key words or concepts used in the manual. Indicate importance of information by placing minor key words under major ones. Give location references for each entry by any one of page number, section number, paragraph number, or illustration number.

###### N+6 Statement of Effort

The statement of effort is a table defining each section of the document, and listing the team members responsible for producing each section. The statement of effort template can be found in Appendix B.

###### N+7 Inspect and Verified by (Quality Assurance)

A note in the footer of the document authorising that the document has been inspected and verified by the quality assurance team.

### 4.3 System Maintenance Document

The System Maintenance Document is prepared for programmers and future maintainers of the system, to specify the roughly detailed architectural structure of the software.

Note - You must update your maintenance document before starting a new sprint.

The System Maintenance Document also includes test provisions that will and have been applied to the software developed. The following table represents a document outline, which should be used as a model for the System Maintenance Document. Each section is composed of a number subsections which address different aspects of the design representation.

|  |
| --- |
| << Front Section + 1.0 Introduction>> |
| 2.0 Software Design Scope  2.1 Major Software Functions  2.2 Major Design Constraints and other Requirements  3.0 Reference Documents  3.1 Existing Software Documentation  3.2 System Documentation  3.3 Vendor Documentation  3.4 Other Documentation  4.0 User Stories  4.1 User Story Dictionary  4.2 Iterative User Story Documents  4.2.1 User Story Definition  4.2.2 Flow of Interaction Diagram  4.2.3 User Story Testing  4.2.4 Integration Test (optional)  4.2.5 Wireframes  5.0 Object-Oriented Design  5.1 High Level System Architecture  5.2 High Level Package Diagram and Components  5.3 Use Case Analysis  5.4 Domain Model and Class Diagram  5.5 Establishment of the Database Objects and Data Access Strategy  5.6 Sequence diagram (optional)  5.7 Object Dictionary  6.0 Software release report  6.1 Usability Test Report  6.2 System (User Story) Test Report  7.0 Additional User Interface Design  7.1 Additional Input and Output Screens  7.2 Additional Data Display Screens  8.0 Special Notes |
| <<Back Section>> |

###### 2.0 Software Design Scope

This section describes the overall scope of the design / maintenance effort. This material is covered in two subsections as follows:

2.1 Major Software Functions

A brief summary is given (possibly in point form) that describes the major software functions that the new system will provide and the benefits these functions provide to the users and the business as a whole.

2.2 Major Design Constraints and other Requirements

This discusses any other constraints or limitations which might affect the design effort. This might include economic constraints (eg: overall budget), scheduling or delivery constraints. Other requirements include the performance, reliability and also software and hardware environments.

###### 3.0 Reference Documents

Specific references to support the design maintenance effort are listed.

3.1 Existing Software Documentation

Any existing documents related to the software being designed will be listed here. At the very least it will include the Project Description document from LMS.

3.2 System Documentation

If the software being developed is to be embedded in an existing system, references to the relevant documentation on that system should be included.

3.3 Vendor Documentation

If the new software is to make use of, or interact with, any third-party software (eg: DBMS, graphics, accounting packages) or unusual hardware (eg: networks, sensors), the relevant documentation provided by the vendors of these products is to be listed.

3.4 Other Documentation

Any other documentation that is relevant to the development of the software design is included in this section.

###### 4.0 User Stories

As one of the most important parts of doing agile project, writing effective User Stories (US) are very crucial. User Stories are written in form of: "As <role/who> <when> <where>, I <what> because <why>."

You must add all user stories in which, role is one of the system actors such as, non-authenticated user, authenticated-user, administrator, operator, etc. The reason is, in agile documentation process you should avoid adding more technical details which are not valuable for the customer. Agile focuses more on coding aspects and such as code clarity during writing it by developers.

4.1 User Story Dictionary

The User Story Dictionary (USD) is a detailed list of the user stories that have been completed up to the end of the current sprint.  These user stories will show the functionality that has been implemented, the benefits they provide to the users, the story points allocated to them and the team members that contributed to the successful completion of the story.

4.2 Iterative User Story Documents

This section is major part of your maintenance document and must be completed ***continuously*** throughout the development lifetime. The user stories in this section must follow the same order as the stories in the dictionary in 4.1.

4.2.1 User Story Definition

This section will contain a detailed description of each user story listed in section 4.1

4.2.2 Flow of Interaction Diagram

The show the more detailed refinement of the identified user interfaces and the flow of data into and out of the system. For those user stories that have more complex interaction between their UI elements, some iFIN diagrams should be used for describing the flow of interaction.

4.2.3 User Story Testing

User story testing describes the tests carried out for one particular user story which might contain a set of tests on functions or methods which are part of that user story and correspond to the acceptance criteria. What you need to document:

1. Acceptance test reports with the below format
2. Your testing codes (or preferably a link to your testing coded which demonstrates test cases and purpose of test)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| User Story | Test ID | Test Objective | Acceptance Criteria | Expected Output | Status (Pass/Fail) |
|  |  |  |  |  |  |

4.2.4 Integration Testing (optional)

Outline the tests, which will be applied to exercise the interfaces between various objects as they are brought together to form the user story. (e.g: between user interface and data objects). Essentially, component test shows that how accurate your user story itself is working (without taking into consideration of interaction of that user story with whole system).

4.2.5 Wireframes

Include any wireframes relevant to the user story. If necessary, highlight any specific relevant features of the wireframes and describe how the wireframes are linked together.

###### 5.0 Object-Oriented Design

This is the process of transformation of the Conceptual Model into the Software Structure Model. The Software Structure Model is a blueprint for the implementation.

5.1 High Level System Architecture

This section will describe a high level architectural design of the system. Describe the various components identified in the system. Give a high-level component design of the system and their relationship to the hardware components.

5.2 High Level Package Diagram and Components

A high level view of the major components to be designed for the system is presented here. Most commonly takes the form of a package diagram. This diagram gives you a more understanding of different components of system and consequently your team and product owner can decide which package/s would be more important for them and more accurate prioritization would be feasible.

5.3 Use Case Analysis

This section will describe the functionality of the system with respect to the input/output data (transformation of data carried out by the system) using use cases. Use case diagram can be delivered in different abstraction levels. You need to draw **one** use case diagram in which major functional requirements can be presented without detailed definition. Apart from mentioned use case diagram, you must prepare use case definition (in form of title, goal, pre-condition, post-condition and etc. ) in detailed for 3 most important use cases in your system.

5.4 Domain Model and Class Diagram

* Choose the most appropriate algorithms and data structures for any given methods and attributes in the objects or classes defined in the Conceptual Modelling stage.
* Define any additional objects or classes, or new internal methods or attributes.

5.5 Establishment of the Database Objects and Data Access Strategy

* Transform the domain objects to database objects
* Develop the Entity Relationship Diagram
* Define data management operations
* If transformation is to Object DBMS, the degree of transformation is low. If it is to Relational DBMS; one needs to map objects and relationships into suitable structures for the relational DB.

5.6 Sequence diagram

Choose at least three of the most valuable user stories and draw detailed sequence diagram in which interactions among different objects, methods are demonstrated. Note that all acceptance criteria must be presented in sequence diagram.

5.7 Object Dictionary

The Object Dictionary is a one-stop reference for all of the objects developed for the system for system designers and maintainers. The object name and description are included, as well as a breakdown of each object’s methods, attributes, and their usage. A sample layout will be provided during the subject to show and ideal format. The object dictionary should be maintained throughout the development of the system. The object dictionary can also be generated through a package like nDocs, or doxygen. You need to update object dictionary as you develop the user stories.

###### 6.0 Software release report

Before releasing your product you need to make sure that it will be ready for deployment in terms of functionality and usability.

6.1 Usability Test Report

Outline the procedure for usability testing; what approach you will take to ensure high usability. Design the necessary questionnaire, etc. Based on usability test report, you should clearly show that what kind of modification has made.

6.2 System Test Report

System testing before release, guarantees all implemented user stories have no defects within the system as a whole.

###### 7.0 Additional User Interface Design

Any interfaces (screens) discovered during the design stage should be added to the maintenance documentation. These may include your user interface designs, input screens or output screens, intermediate data screens, warning screens, error messages or help windows. You should specify format, colour, and function keys used.

7.1 Additional Input and Output Screens

Any new screens identified facilitating the input or output of data. This includes revisions to controls on screens as well, e.g. list boxes are now used to input data instead of text boxes.

7.2 Additional Data Display Screens

Any new screens identified which communicate important information with the user. This may include newly detected error screens, new format for the help screens, etc.

###### 8.0 Special Notes

Include any other relevant matters not included elsewhere. This might include special algorithms descriptions, alternative procedures, etc.

### 4.4 Reflection Component for Masters Students

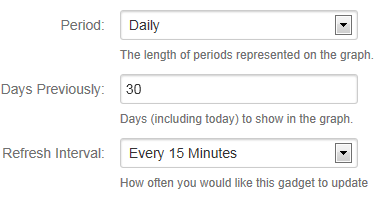
#### 4.4.1 One time Setup

Create jql search filters for the following:

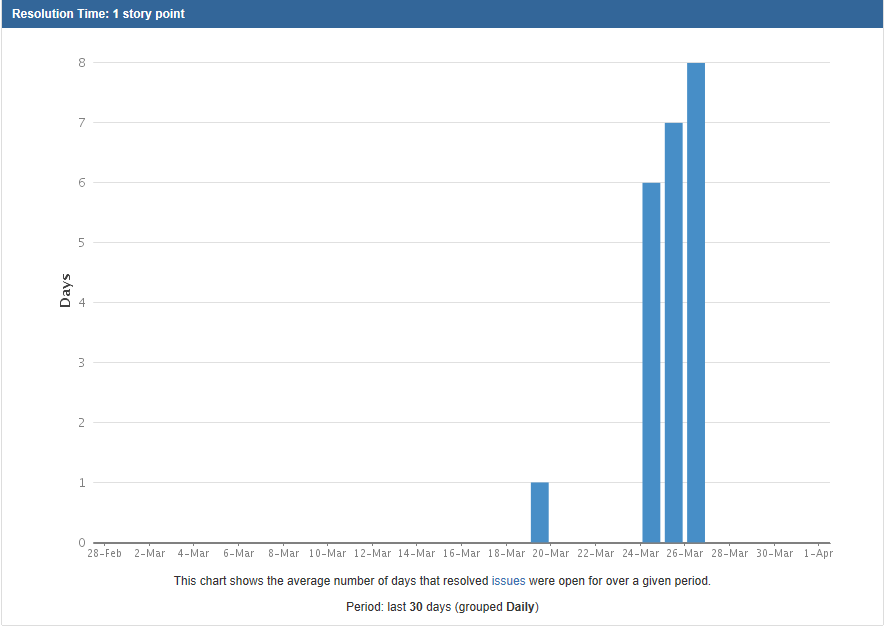
1. All tasks (Epics, stories, tasks and subtasks) assigned to you with story points of 2 or higher
2. All epics created in the previous sprint (think about which date you should use)
3. All stories and tasks currently in progress but blocked in the current sprint (In progress means out of to-do but not in done)
4. All finished tasks from previous month of all types that have:

* 1 story point
* 2 story points
* 3 story points
* 5 story points

Create a new dashboard for reporting and share it only with Project PRA/B (supervisors). Add a “Resolution Time” gadget for each of the queries created in section (i) above. Use the following settings:



You should get a graph similar to this:



#### 4.4.2 Sprint Reports

At the end of each sprint masters students are required to submit a document reflecting on how their team progressed and performed through their sprint.

##### Task 1: Average Resolution Time

|  |  |
| --- | --- |
| Story Points/ T-Shirt Size | Days |
| 1/S |  |
| 2/M |  |
| 3/L |  |
| 5/XL |  |

Calculate average number of days from each graph and fill in the table below with your averages.

##### 

##### Task 2: Trend Reports

Your task is to describe and reflect on trends and observations after every sprint from the following two charts:

1. Burndown chart
2. Velocity chart  
   Follow the process in the table below for your reflection.

|  |
| --- |
| RECORD (50 words minimum) *Describe / re-tell key elements of what you’ve learned or experienced while going through your teams sprint.* |
|  |
| RELATE (50 words minimum) *Draw a relationship between your current position or understandings and identify aspects of the obsesrvations which have personal meaning or connect with your experience.* |
|  |
| REASON (50 words minimum) *Explore the relationship between theory and practice and seek a deep understanding of why something has happened throughout the last sprint. You can explore or analyse a concept, event or experience by asking questions and looking for answers, reviewing the stories considering alternatives and multiple perspectives.* |
|  |
| RECONSTRUCT (50 words minimum) *Discuss improvements which could be made or identify something you need or plan to do or change. You should be able to generalise and/or apply learning to other contexts and future professional practice. This may involve developing general principles or formulating personal theories on how to improve upon poor performance. How would I deal with this next time? What might work and why? Are there different options? Can I make changes that will benefit not only me but also the team?* |
|  |

### 4.5 User Documentation

The User Documentation is prepared for the end-user of the software, specifically to provide a variety of information required to operate the software. The User Document is used as a supplement to the help functionality provided in the software. An introductory kit gives ‘getting started’ advice on using the system, followed by a comprehensive guide on each of the system functionalities and how to use the system to perform the functionalities.

|  |
| --- |
| << Front Section + 1.0 Introduction>> |
| 2.0 Introductory Kit  3.0 Getting Started Guide  4.0 User Manual  5.0 System Administration Guide  6.0 Installation Guide |
| <<Back Section>> |

###### 2.0 Introductory Kit

The Introductory Kit should begin with a summary of features of the system. It should explain the functionality provided by the system to the potential user.

###### 3.0 Getting Started Guide

Quickly explain, in simple terms, how to perform the main uses cases of the system. This should be liberally illustrated with examples and may take the form of a step-by-step tutorial. Diagrams/illustrations are very useful. Describe how to initiate the system and how the user might make use of the common system facilities.

###### 4.0 User Manual

The user manual is a *definitive reference* on system usage. Each sub-section should illustrate in simple terms how to use a specific function. It should be *complete*, and must go *systematically* through each possible action, showing every menu and screen. Wherever possible try to standardise the presentation format. This section should also present, possibly in tabular form, the error reports generated by the system, and the appropriate user response for different errors.

###### 5.0 System Administration Guide

For systems that require operator intervention, a system administration guide must be provided. This should describe the messages generated at the system console and how to react to these messages. It should also include such items as backup of user data, file management, recovery from a system crash, and procedure to bring the system to the state before the system crash. If system hardware is involved, it might also explain the Administrator’s task in maintaining that hardware.

###### 6.0 Installation Guide

Explain, step by step, how to install the system and tailor it for particular hardware configurations. This should include a description of the machine-readable media on which the software is supplied (its format, character codes used) and the files making up the system. It should then describe the minimal hardware configuration required, and the configuration dependent files that must be changed in order to tailor the system to a particular host system.

## Chapter 5 – Assessment Guidelines for Subject A

### 5.1 Introduction

This section covers the guidelines for assessment of all the meetings, presentations and submissions in subject A. Please read the following pages carefully before each assessable submission commences. Make sure you address all the stated criteria in all of your software deliverables. Pay attention to planning, team dynamics, and follow carefully software engineering best practices, such as usability, reusability, configuration management, and component-based development. Make sure you update the schedule and scope of the project regularly to incorporate the changes and stage of the development.

### 5.2 Assessment Components

The work carried out throughout the semester is designed for continuous assessment of the individual’s contribution to the Industry Project. There are three types of assessment for the work carried out by a team member: (i) Sprint meeting assessments, (ii) Assessment of presentations and (iii) Assessment of major submissions.

The teams are encouraged to use so called the 4R approach for their weekly contribution to the development and documentation of the system. The teams are encourage to use the design document as a reference and follow the 4R approach:

* Role: Each team member will assume a role as chief designer, coder, documenter, tester, administrator, etc. He/she will still contribute to other aspects.
* Responsibility: He/she will be responsible for standard/template, allocation, quality, and correctness of that aspect of the project.
* Revision: He/she will be responsible to collate, check quality, update and revise that aspect of project
* Report: He/she will be responsible to report to team and supervisor and in presentations that aspect of the project.

Students in this subject will be informed of the result of individual and team assessments throughout the subject, however the mark reported for the subject A will be XC (continue). Students in this subject have to gain at least 50% mark for all assessments so they can continue to subject B in following semester. The final mark for the subject will be the average mark for subject A and subject B.

#### 5.2.1 Assessment Subject A

|  |  |  |  |
| --- | --- | --- | --- |
| Assessment | % of final mark | | Submission Date |
|  | **3PRA** | **5PRA** |  |
| Sprint Assessments. Sections 5.5 5.6 (and 5.7 for Masters students) of this Handbook(Individual) | 35 | 25 | Weeks 3,5, 7, 9, 11 |
| Phase 1 Presentation(group) | 10 | 10 | Week 8 |
| Phase 2 Submission: Section 4.3  (distributed based on sprint assessment) | 35 | 35 | Week 13 |
| Phase 2 Presentation(group) | 20 | 20 | Week 13 |
| Masters Reflection (individual) | 0 | 10 | Weeks 3, 5, 7, 9, 11 |

Penalties are applied to late assignments. If there are circumstances that prevent the assignment being submitted on time, an application for special consideration may be made. See the section on Special Consideration below. Students must plan their work to allow for both scheduled and unscheduled downtime.

#### 5.2.2 Requirements for continuing to subject B:

To obtain a satisfactory mark for continuing to subject B in the second semester, you must:

1. accumulate at least 50% overall forms of assessment

#### 5.2.3 Feedback on Assessment

Assignments will be returned within two weeks of the last day of assignment submission. Announcements regarding when and where assignments may be collected from will be made in lectures and on the subject LMS. The assessment result for team progress and presentations will be published on subject LMS within 2 weeks.

#### 5.2.4 Special Consideration

The School of Engineering and Mathematical Sciences (SEMS) applies the standard University policies and procedures for special consideration. If you believe your performance in an assessment has been adversely affected by an illness or other serious circumstances, submit an application to the Student Centre in a timely manner accompanied by supporting documentation. Each application is considered (special consideration) by the subject coordinator. A further assessment (special assessment) is not automatically granted upon receipt of an application. Special consideration should be submitted here: <http://www.latrobe.edu.au/students/admin/forms/special-consideration>

Further, this only applies to assessments worth 15% or greater, lower it is up to the lecturers discretion.

### 5.3 Major Submission Assessments for Project A

#### 5.3.1 System Maintenance Document

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Assessment Criteria | | | | | | | | | | | | | | Comments / Marks | | | |
| General Presentation:  Documents Titled and Pages Properly Linked/Numbered  Project & Team Identification  Table of Contents  Index & Glossary | | | | | | | | | | | | | | ( /2) | | | |
| Scope  Major Software Functions  Major Design Constraints, Limitations | | | | | | | | | | | | | | ( /3) | | | |
| References  Existing document, System document  Vendor document | | | | | | | | | | | | | | ( /3) | | | |
| Object Oriented Design (32) |  | | | Clarity | | Correctness | | | Standards | | Detail | | Comments | | | |
| High Level Architecture Design (4) | | | (1) | | (1) | | | (1) | | (1) | |  | | | |
| Component / Container / Class structure and their communication (8) | | | (2) | | (2) | | | (2) | | (2) | |
| ERD Diagram and data access strategy (8) | | | (2) | | (2) | | | (2) | | (2) | |
| Sequence Diagrams for implemented User Stories (8) | | | (2) | | (2) | | | (2) | | (2) | |
| Methods Reference (4) | | | (1) | | (1) | | | (1) | | (1) | |
| Each diagram is evaluated by four criteria: correctness, clarity of expression, compliance with standards, and detail. Correctness refers to whether there are any logical errors and inconsistencies in the diagrams presented. Are incorrect relationships being used? Does the diagram accurately represent information? Clarity of expressions refers to how well the diagram is communicated. Are the fonts legible? Are the diagrams cluttered? Are the explanations coherent? Compliance with standards refers to whether or not a diagram complies with standard notations. Is the diagram valid UML? Are the appropriate connector types used? Is the diagram’s notation made up? Detail refers to how fully the view of the system is represented.  Each of these dimensions will be marked as either good, satisfactory, or unsatisfactory which correspond to two, one, and zero marks. | | | | | | | | | | | | | | | |
| User Stories (44) |  | | | | Clarity | | | Correctness | | Standards | | Detail | | | | Comments | |
| User Story Dictionary (4) | | | | (1) | | | (1) | | (1) | | (1) | | | |  | |
| Prototype screens for entire system and Responsiveness design (16) | | | | (4) | | | (4) | | (4) | | (4) | | | |
| User Story Definition (16) | | | | (4) | | | (4) | | (4) | | (4) | | | |
| Flow of Interaction (8) | | | | (2) | | | (2) | | (2) | | (2) | | | |
| Each User Story is evaluated by four criteria: correctness, clarity of expression, compliance with standards, and detail. Correctness refers to whether all is defined precisely and correctly. Standard denotes user story definition; diagrams are following defined standard and structure. Your definition must be detailed enough. As an example, have you covered all acceptance criteria or all aspects in your prototype? Each of these dimensions will be marked as either good, satisfactory, or unsatisfactory. | | | | | | | | | | | | | | | | |
|  | | | Testing Method | | | | Effectiveness | | | | | | | | Coverage | | |
| Software Release Report (15) | | **Usability Test Report (5)** | The tests are precisely defined and the aim and method of the test are appropriately defined  Must include the audience description, metrics definition, approach/plan (might include survey with questions to ask, should include observations of users performing tasks with UI) (2) | | | | Usability Test Report clearly shows how your test has improved your product in terms of usability. A good before-after comparison should be included. (2) | | | | | | | | There are enough observations for all user interface components and their associated functionalities (1) | | |
| **User Story Testing for implemented user stories (5)** | The testing clearly defines the steps required for the user stories, the inputs for each user story, the expected results and pass/fail criteria are clearly and precisely defined. The intended testing procedure is appropriate and clearly defined. (2) | | | | The test cases show that the system operates correctly through the main flow and appropriately handles alternate flows. (1) | | | | | | | | There are test cases for all operations and states for all implemented user story.  (2) | | |
| **Unit Testing (5)** | An appropriate unit testing framework has been selected. Unit tests are clearly defined for each relevant function. (2) | | | | The test cases show that the system operates correctly through the main flow and appropriately handles alternate flows. (1) | | | | | | | | There are test cases for all operations and states for all implemented and relevant functions. (2) | | |
| Special Notes (optional)  Appendix | | | | | | | | | | | | | | | ( /1) | | |

### 5.4 Major Meetings and Presentations Assessments

#### 5.4.1 Phase 1 Demonstration Assessment

You have spent considerable time covering tasks or activities associated with the initial stages of a development project. The ground gained so far is familiar to you and to the users who have been involved with the project for as long as you have, but it is new to other people in the organisation, such as Management. To make the Management in your organisation aware of your project activities, you have decided to give a short presentation on their progress on user stories implementation. (Submit the presentation file).

PRA – Week 8 – 20 Minutes + Questions Team Letter: Date: / /2019 Marker:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Excellent | Good | Satisfactory | Unsatisfactory |
| Punctuality (5) | The team has tested the room’s setup in the days before. Some team members arrive early to setup the room and the rest arrive on time. (5) | Some team members arrive early to setup the room, but there are some technical issues; the rest arrive on time. (3) | All team members arrive on time. There are technical issues. (1) | Some team members are late or absent without an apology. (0) |
| Presentation Purpose (5) | The team clearly states the purpose of the presentation and explains what material will be covered and why. (5) | The team clearly states the purpose of the presentation and lists the material that will be covered. (3) | The team mentions the purpose of the presentation and lists some of the material they will cover. (1) | The team makes no mention of the purpose of the presentation or the material that will be covered. (0) |
| System Overview (10) | The team correctly and accurately identifies all of the benefits the current iteration of the system provides to the end users. They also identify any client defined constraints they have to work within. (10) | The team correctly lists most of the benefits the current iteration of the system provides, and any client stipulated constraints that they must work within. (7) | The team lists some of the benefits of the current iteration of the system provides. Some of the descriptions are vague or incomprehensible and may not match exactly with the client's intents. (3) | The team incorrectly lists the benefits of the current iteration of the system provides or fails to mention anything at all. (0) |
| User Stories (10) | The team clearly shows all key user stories. They describe how they prioritised the user stories. They show the acceptance criteria, story points, and the story format. (10) | The team identifies a few of the key user stories. There’s little prioritisation or it is questionable. They show some of the acceptance criteria, story points and the story format. (7) | The team gives a vague description of the key user stories. They enumerate some user stories. There are irrelevant use stories and major omissions. Little in terms of priorities, acceptance criteria, story points and story format is shown. (3) | The team ignores the user stories completely or gives an incomprehensible description. (0) |
| Database & Data Interfaces (10) | The team completely and clearly describes the ERD and schema for the system in all *three* areas: correct diagram, correct content and correct syntax. (10) | The team completely and clearly describes the ERD and schema for the system in *two* of the areas: correct diagram, correct content and correct syntax. (7) | The team completely and clearly describes the ERD and schema for the system in *one* of the areas: correct diagram, correct content and correct syntax. (3) | The team fails to describe the ERD and schema for the system in *any* of the areas: correct diagram, correct content and correct syntax. (0) |
| Architecture Diagram (10) | The team presents the solution’s high level architecture clearly showing the system boundary, all of the major software components, devices, users, and interfaces with external systems. The diagram is clear, concise and logical. They satisfy all *three* areas: correct diagram, correct content and correct syntax. (10) | The team presents the solution’s high level architecture clearly showing the system boundary, some of the major software components, devices, users, and interfaces with external systems. The diagram is clear, concise and logical. They satisfy *two* areas: correct diagram, correct content and correct syntax. (7) | The team presents the solution’s high level architecture showing some of the major software components, devices, users, and interfaces with external systems. The diagram is vague and contains minor logical errors and omissions. They satisfy *one* areas: correct diagram, correct content and correct syntax. (3) | The team presents the solution’s high level architecture. It is illogical and incomplete. They satisfy *none* of the areas: correct diagram, correct content and correct syntax. (0) |
| UI Prototype (20) | The UI prototype clearly communicates a complete description of the software (small & big screens) and its interaction with the user. The team shows, in detail, how some of the important user stories will work. The prototype elicits, refines and/or validates requirements from the client. (20) | The UI prototype clearly communicates a partial description of the software (small & big screens) and its interaction with the user. The team shows how some of the use stories can be completed with the prototype. (14) | The UI prototype communicates a partial description of the software (small & big screens) and its interaction with the user. The team indicates how some of the use stories might be completed with the prototype. (7) | The team does not appear to satisfy any of the project's goals. The UI prototype is deeply flawed and no user interaction is shown. (0) |
| Sample Implementation (10) | The team shows sample implementation for all *three* of the following: item 1, item 2 and item 3 (should be discussed with the supervisor). (10) | The team shows sample implementation for all *two* of the following: item 1, item 2 and item 3. (7) | The team shows sample implementation for all *one* of the following: item 1, item 2 and item 3. (3) | The team shows no implementation. (0) |
| Presentation (10) | The topics of the presentation followed logically from one to another. Changes between presenters are done in an efficient and appropriate manner. All team members present. The presentation appeared well rehearsed without any technical issues. (10) | The topics of the presentation followed logically from one to another most of the time. Most of the time, changes between presenters are done in an efficient and appropriate manner. Most team members present. (7) | The topics of the presentation followed logically from one to another some of the time. There were some unnecessary changes between presenters and minor technical issues. Only few team members present. (3) | There were constant technical issues. The presentation kept stopping to fix problems. The presentation was disjointed. Only few team members present. (0) |
| Conclusion (5) | The team explains how this presentation will feed into the development process and what the next presentation will be about in terms of the project. (5) | A minor mention is made to future development *and* the next presentation. (3) | A minor mention is made to future development *or* the next presentation. (1) | No mention is made to future development or the next presentation. (0) |
| Question Time + Feedback (2.5) | The team is able to concisely and clearly answer all of the questions asked by the markers and client.  The team takes note of all of the feedback given. When appropriate, they ask for clarification. (2.5) | The team is able to clearly answer most of the questions asked by the markers and client. The team takes note of all of the feedback given. (1.5) | The team is able to answer some of the questions asked by the markers and client. The team does not take note of the feedback given. (1) | The team gives incomprehensible answers to questions asked by the client and markers. The team does not take note of the feedback given. (0) |
| Etiquette (2.5) | The presenters conducted themselves in a professional manner at all times. (2.5) | The presenters were rude, disrespectful, impatient, and/or gave inappropriate criticism. (0) |  |  |
| Total: |  |  |  |  |

#### 5.4.2 Phase 2 Demonstration Assessment

You have spent considerable time covering tasks or activities associated with the initial stages of a development project. The ground gained so far is familiar to you and to the users who have been involved with the project for as long as you have, but it is new to other people in the organisation, such as Management. To make the Management in your organisation aware of your project activities, you have decided to give a short presentation on their progress on user stories implementation. (Submit the presentation file).

PRA – Week 12 – 30 Minutes + Questions Team Letter: Date: / /2019 Marker:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Excellent | Good | Satisfactory | Unsatisfactory |
| Punctuality (5) | Some team members arrive early to setup the room and the rest arrive on time. (5) | All team members arrive on time. (2.5) |  | Some team members are late or absent without an apology. (0) |
| Presentation Purpose (5) | The team clearly states the purpose of the presentation and explains what material will be covered and why.(5) | The team clearly states the purpose of the presentation and lists the material that will be covered. (3) | The team mentions the purpose of the presentation and lists some of the material they will cover. (1.5) | The team makes no mention of the purpose of the presentation or the material that will be covered. (0) |
| System Overview (5) | The team correctly and accurately identifies all of the benefits the current iteration of the system provides to the end users. They also identify any client defined constraints they have to work within. (5) | The team correctly lists most of the benefits the current iteration of the system provides, and any client stipulated constraints that they must work within. (3) | The team correctly lists most of the main benefits the current iteration of the system provides. Some of the descriptions are vague or incomprehensible and may not match exactly with client's intents. (1.5) | The team incorrectly lists the benefits the current iteration of the system provides or fails to mention anything at all. (0) |
| Sprint Planning (5) | The team correctly identifies all of the project backlog and select clear plan for the sprint solution. (5) | The team identifies some of the project backlog and select some plan for the sprint solution. (3) | The team identifies little of backlog and select not clear plan for the sprint solution. (1.5) | The team does not demonstrate any part of their project plan. (0) |
| Sprint User Stories (5) | The team clearly show some of the stories they have selected for their last sprint only. They describe how they got to each conclusion.  They show the acceptance criteria, story points and the story format. (5) | The team identifies few of the stories they have selected for their last sprint only. They some of reasons how they got to each conclusion.  They show some of the acceptance criteria, story points and the story format. (3) | The team gives a vague description of the problem domain. They enumerate some user stories. There are irrelevant use stories and major omissions. (1.5) | The team ignores the user stories completely or gives an incomprehensible description. (0) |
| Structural View (5) | The team completely and clearly describes the structural view of the problem domain, its entities, attributes and, relationships. The team gives a rationale for the inclusion of entities into the problem domain. (5) | The team clearly describes the structural view of the problem domain, its entities, attributes and, relationships. There are some minor omissions and some of the entities are given vague descriptions. (3) | The team gives a vague and incomplete description of the entities and their relationships. There are some irrelevant entities and minor omissions. (1.5) | The team gives an incomprehensible description of the structural view of the problem domain. |
| Standards in system modelling (5) | The team appropriately selects and correctly follows standard modelling techniques in all work.(5) | The majority of the work correctly follows standard modelling techniques. (3) | Some of the work correctly follows standard modelling techniques. (1.5) | It is not apparent that any standards are being followed what so ever. |
| Architecture (5) | The team presents the solution’s high level architecture clearly showing the system boundary, all of the major software components, devices, users, and interfaces with external systems. The diagram is clear, concise and logical.(5) | The team presents the solution’s high level architecture clearly showing the system boundary, some of the major software components, devices, users, and interfaces with external systems. The diagram is clear, concise and logical. (7) | The team presents the solution’s high level architecture showing some of the major software components, devices, users, and interfaces with external systems. The diagram is vague and contains minor logical errors and omissions. (3) | The team presents the solution’s high level architecture. It is illogical and incomplete. |
| Sprint Implementation (35) | The sprint implementation clearly communicates a complete description of the outwardly visible part of the system. The team shows the implementation of user stories described earlier including:  - Connectivity of the complete system (even if just as a prototype / design) (5)  - Database (schema created) & logic (10), user interface (10), scenarios to test functionality (5)  - (Dummy) data populated (5)  \* Teams don't have to show the items mentioned above separately. They should be demonstrated by the execution of the functionality of your system  \*\* If it’s not possible to have sections/ features fully functional, there has to be a clear design on how everything will be implemented/ integrated later on | The sprint implementation clearly communicates a partial description of the outwardly visible part of the system. The team shows how some of the use stories can be completed with the wireframes. (25) | The prototype communicates a partial description of the outwardly visible part of the system. The team indicates how some of the use stories might be completed with the wireframes. (15) | The team does not appear to satisfy any of the project's goals. |
| Agile Maturity (5) | The team clearly understands and reflects on areas that the team can improve. The team attempts to use metrics to demonstrate and compare improvement. The team put clear action plans to improve on Agile process. (5) | The team identifies and reflects on some areas that the team can improve. The team attempts to demonstrate and compare improvement. The team put some plans to improve on Agile process. | The team identifies the need for Agile process improvement. The team understand the need to put plans to improve on their process. | No Agile process improvement. |
| Presentation (5) | The topics of the presentation followed logically from one to another. Changes between presenters are done in an efficient and appropriate manner. The presentation appeared well rehearsed without any technical issues. (5) | The topics of the presentation followed logically from one to another most of the time. Most of the time, changes between presenters are done in an efficient and appropriate manner. (3) | The topics of the presentation followed logically from one to another some of the time. There were some unnecessary changes between presenters and minor technical issues. (1.5) | There were constant technical issues. The presentation kept stopping to fix problems. The presentation was disjointed. |
| Conclusion (5) | The interviewer explains how this presentation will feed into the development process and what the next presentation will be about in terms of the project.(5) | A minor mention is made to the next meeting and what they can expect. (2.5) | No mention is made to the next meeting. |  |
| Question Time + Feedback (5) | The team is able to concisely and clearly answer all of the questions asked by the markers and client.  The team takes note of all of the feedback given. When appropriate, they ask for clarification. (5) | The team is able to clearly answer most of the questions asked by the markers and client. The team takes note of all of the feedback given. (3) | The team is able to answer some of the questions asked by the markers and client. The team takes note of all of the feedback given. (1.5) | The team gives incomprehensible answers to questions asked by the client and markers. |
| Etiquette (5) | The presenters conducted themselves in a professional manner at all times. (5) | | | The presenters were rude, disrespectful, impatient and gave inappropriate criticism. |

5.5 Reflection Assessment for Masters Students

This is worth 10% of Masters Students Sprint Review/Retrospective assessments. Refer to Section 4.4 – Reflection Component for Masters Students.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Excellent | Satisfactory | Unsatisfactory |
| Section A:  Story Point Average Calculation | Personal Dashboard fully maintained. (1)  All averages for each Story size in the sprint are calculated. (3) | Personal Dashboard partly maintained  Some averages for each story size are calculated. (1.5) | No Personal dashboard i (0)  No averages are calculated. (0) |
| Section B:  Reflection | Reflection is coherent, succinct and well-reasoned. (3)  Appropriate agile charts/performance measures are referred to. (3) | Reflection is somewhat succinct and reasoned. (1.5)  Some agile charts/performance measures are referred to. (1.5) | Written in dot point form. (0)  No reasoning. (0)  No agile charts are mentioned. (0) |

Major submissions or presentations would reflect continuous and collective planning, research, development, careful review, and documentation of work carried out by team members. It would mirror the individual team members' weekly contribution prior to a submission. The average weekly contribution of each team member during the weeks prior to a submission is considered as the “individual factor” toward the submission.

Each major submission will be assessed as one group submission, based on the individual contribution in terms of quantity, quality, consistency and history of tasks carried out by team members toward such submission. This assessment mark will be proportioned to each team member based on the “individual factor” of contribution to such submission. Individual mark for each major submission will be calculated as following:

<individual factor> is calculated based on sprint marks during weeks prior to submission, <average factor> is the average of “individual factors” for all team members.

For example for a team mark (e.g. for a major submission) of 77.5, the mark distribution is shown in the table below. Hence the weekly individual contributions have impact to the final mark: once directly as part of the mark and once as a factor for distribution of the team marks.

|  |  |  |
| --- | --- | --- |
| Team Member | Factor | Mark |
| 1 | 64.2 | 77.0 |
| 2 | 73.8 | 88.5 |
| 3 | 71.8 | 86.1 |
| 4 | 78.4 | 94.0 |
| 5 | 53.8 | 64.5 |
| 6 | 45.8 | 54.9 |

|  |  |  |  |
| --- | --- | --- | --- |
| Average Factor 🡪 | **64.6** | **77.5** | 🡨 **Team Mark** |

## Chapter 6 – Assessment Guidelines for Subject B(TBA)

This section covers the guidelines for assessment of all the meetings, presentations and submissions in subject A. Please read the following pages carefully before each assessable submission commences. Make sure you address all the stated criteria in all of your software deliverables. Pay attention to planning, team dynamics, and follow carefully software engineering best practices, such as usability, reusability, configuration management, and component-based development. Make sure you update the schedule and scope of the project regularly to incorporate the changes and stage of the development. The following table lists what you need to submit and when. The assessment criteria for each submission can be found throughout this chapter.

|  |  |  |  |
| --- | --- | --- | --- |
| Item | Due Date | % | Weighted |
| 1. Final System presentation 45-minutes (team) | Week 11 | 20 | No |
| 2. Demo 30-minutes (team) + Poster | Week 5 | 15 | No |
| 4. User Guide Submission (15%), System Maintenance Document Submission (10%), and Source Code Submission (5%) (team) | Week 11 | 35 | Yes |
| 5. Sprint meeting report, includes individual and team progress using agile tools | Weeks 2, 4, 6, 8, 10 | 30 | No |

Penalties are applied to late assignments (5% of total possible marks for that task is deducted per day, accepted up to 5 days after the due date only). If there are circumstances that prevent the assignment being submitted on time, an application for special consideration may be made. See the section on Special Consideration above. Note that delays caused by computer downtime cannot be accepted as a valid reason for a late submission without penalty. Students must plan their work to allow for both scheduled and unscheduled downtime.

### Sprint Review/Retrospective Meeting Assessment – 30%

Team Name: Date:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Description | Excellent | Acceptable | Not Acceptable | |
| General Meeting Operation (6) |  |  |  | |
| The chair person and secretary are listed on the agenda |  | 1 | 0 | |
| Agenda items are relevant and appropriately time boxed |  | 1 | 0 | |
| All discussions related to the agenda |  | 1 | 0 | |
| All the agenda points are covered |  | 1 | 0 | |
| The minutes from the last sprint are confirmed\* |  | 1 | 0 | |
| The meeting finished on time |  | 1 | 0 | |
|  |  |  |  | |
| Sprint Review (12) |  |  |  | |
| The team defined tasks with an appropriate level of detail and story points | 2 | 1 | 0 | |
| The team scheduled an appropriate number of tasks | 2 | 1 | 0 | |
| Each member took an appropriate amount of tasks | 2 | 1 | 0 | |
| The team completed an appropriate number of tasks | 2 | 1 | 0 | |
| The team produced quality work as determined by metrics, standards or group feedback | 2 | 1 | 0 | |
| The outputs of the team are integrated and stored centrally | 2 | 1 | 0 | |
| *notes* | | | | |
| Sprint Retro (12) |  |  |  | |
| The team accurately identified activities, processes, behaviours and tools that helped progress | 3 | 1.5 | 0 | |
| The team accurately identified the activities, processes, behaviours and tools that slowed down progress | 3 | 1.5 | 0 | |
| The team identified actionable steps to improve the next sprint | 3 | 1.5 | 0 | |
| The team updated activities, processes, behaviours and tools to improve progress. | 3 | 1.5 | 0 | |
| *notes* | | | | |
| Total: | | | |  |

### Sprint Review/Retrospective Individual Assessment – 70%

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1. Student Name |  |  |  |  |  |
| General Meeting Operation (20) |  |  |  |  |  |
| The individual contributed to discussions | 10 5 0 | 10 5 0 | 10 5 0 | 10 5 0 | 10 5 0 |
| The individual takes detailed notes on their responsibilities in the upcoming sprint | 10 5 0 | 10 5 0 | 10 5 0 | 10 5 0 | 10 5 0 |
|  |  |  |  |  |  |
| Sprint Review (35)  *notes*  *🗶 tasks accepted but not finished*  *✓ tasks finished* |  |  |  |  |  |
| The individual took an appropriate number of tasks (deduct marks for taking too many tasks) | 5 2.5 0 | 5 2.5 0 | 5 2.5 0 | 5 2.5 0 | 5 2.5 0 |
| The individual provided evidence for completed tasks and/or valid reasons for not completing tasks | 10 5 0 | 10 5 0 | 10 5 0 | 10 5 0 | 10 5 0 |
| The individual produced quality work as determined by metrics, standards or group feedback | 10 5 0 | 10 5 0 | 10 5 0 | 10 5 0 | 10 5 0 |
| The individual shows evidence of incorporating the last 2 workshops into their project | 10 5 0 | 10 5 0 | 10 5 0 | 10 5 0 | 10 5 0 |
|  |  |  |  |  |  |
| Sprint Retro (15) |  |  |  |  |  |
| The individual maintains a good working relationship with the other team members and/or manages conflicts constructively | 10 5 0 | 10 5 0 | 10 5 0 | 10 5 0 | 10 5 0 |
| The individual follows team processes such as scrum on JIRA, communication channels, version control | 5 2.5 0 | 5 2.5 0 | 5 2.5 0 | 5 2.5 0 | 5 2.5 0 |
|  |  |  |  |  |  |
| Team Total: Individual Total: |  |  |  |  |  |
| Total: |  |  |  |  |  |

### 6.3 Reflection Assessment for Masters Students

This is worth 10% of Masters Students Sprint Review/Retrospective assessments. Refer to Section 4.4 – Reflection Component for Masters Students.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Excellent | Satisfactory | Unsatisfactory |
| Section A:  Story Point Average Calculation | Personal Dashboard fully maintained. (1)  All averages for each Story size in the sprint are calculated. (3) | Personal Dashboard partly maintained  Some averages for each story size are calculated. (1.5) | No Personal dashboard i (0)  No averages are calculated. (0) |
| Section B:  Reflection | Reflection is coherent, succinct and well-reasoned. (3)  Appropriate agile charts/performance measures are referred to. (3) | Reflection is somewhat succinct and reasoned. (1.5)  Some agile charts/performance measures are referred to. (1.5) | Written in dot point form. (0)  No reasoning. (0)  No agile charts are mentioned. (0) |

### 6.4 Demonstration Assessment

PRB – Week 5 Team Letter: Date: /08/2019 Marker:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Excellent | Good | Satisfactory | Unsatisfactory |
| Punctuality (5) | Some team members arrive early to setup the room and the rest arrive on time. (5) | All team members arrive on time. (2.5) |  | Some team members are late or absent without an apology. (0) |
| Presentation Purpose (5) | The team clearly states why they are there today. What they hope to get out of the presentation and what they’re going to cover in this presentation. (5) | The team clearly states the purpose of the presentation and lists the material that will be covered. (3) | The team lists some of the material they will cover. (1.5) | The team makes no mention of the purpose of the presentation or the material that will be covered. (0) |
| System Overview (5) | The team correctly and accurately identifies all of the benefits the current iteration of the system provides to the end users. They also identify any client defined constraints they have to work within. (5) | The team correctly lists most of the benefits the current iteration of the system provides, and any client stipulated constraints that they must work within. (3) | The team correctly lists most of the main benefits the current iteration of the system provides. Some of the descriptions are vague or incomprehensible and may not match exactly with client's intents. (1.5) | The team incorrectly lists the benefits the current iteration of the system provides or fails to mention anything at all. (0) |
| Sprint Planning (5) | The team shows the tasks that they have allocated to their current sprint. They explain how they are working towards being able to successfully deliver the next iteration of the system by the end of this sprint. (5) | The team shows some of the tasks they are currently working on. They do not go into much detail about these tasks. (3) | The team lists their current tasks. (1.5) | The team does not demonstrate any part of their project plan. (0) |
| Sprint User Stories (5) | The team clearly show some of the stories they have selected for their last sprint only. They describe how they got to each conclusion. The team also demonstrates how they have taken previous feedback on board.  They show the acceptance criteria, story points and the story format. (5) | The team identifies few of the stories they have selected for their last sprint only. They some of reasons how they got to each conclusion.  They show some of the acceptance criteria, story points and the story format. (3) | The team gives a vague description of the problem domain. They enumerate some user stories. There are irrelevant use stories and major omissions. (1.5) | The team ignores the user stories completely or gives an incomprehensible description. (0) |
| Structural View (5) | The team completely and clearly describes the structural view of the problem domain, its entities, attributes and, relationships. The team gives a rationale for the inclusion of entities into the problem domain. (5) | The team clearly describes the structural view of the problem domain, its entities, attributes and, relationships. There are some minor omissions and some of the entities are given vague descriptions. (3) | The team gives a vague and incomplete description of the entities and their relationships. There are some irrelevant entities and minor omissions. (1.5) | The team gives an incomprehensible description of the structural view of the problem domain. |

PRB – Week 5 Team Letter: Date: /08/2019 Marker:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Excellent | Good | Satisfactory | Unsatisfactory |
| Punctuality (2.5) | The team has tested the room’s setup in the days before. Some team members arrive early to setup the room and the rest arrive on time. (2.5) | Some team members arrive early to setup the room but there are some technical issues; the rest arrive on time. (1.5) | All team members arrive on time. There are technical issues. (1) | Some team members are late or absent without an apology. (0) |
| System Overview (2.5) | The team correctly and accurately identifies all of the benefits the current iteration of the system provides to the end users. They also identify any client defined constraints they have to work within. (2.5) | The team correctly lists most of the benefits the current iteration of the system provides, and any client stipulated constraints that they must work within. (1.5) | The team lists some of the benefits the current iteration of the system provides. Some of the descriptions are vague or incomprehensible and may not match exactly with client's intents. (1) | The team incorrectly lists the benefits the current iteration of the system provides or fails to mention anything at all. (0) |
| Modifications and Improvements in System Design (5) | The Team list all the required changes/improvements in their design (DB Schema, System Architecture, UI, etc…) since the last presentation, and justifies the reasons for these changes.(5) | The Team list part of the changes/improvements in their design (DB Schema, System Architecture, UI, etc…) since your last presentation, but the justifications is not clear.(3) | The Team list part of the required changes/ improvements in their design (DB Schema, System Architecture, UI, etc…) since your last presentation, and the justifications is not clear.(1) | The Team incorrectly list part of the required changes in their design (DB Schema, System Architecture, UI, etc…) since your last presentation, and does not provide any justifications (0) |  |
| Implementation (50) | The team shows the implementation. Team should demo major functionality of the system. Please contact your supervisor to be informed which user stories/functionalities needs to be demonstrated. (50) | The team shows the implementation of some key user stories. (25) | The team shows a vague or questionable implementation of few key user stories (15) | The team does not show any implementation or can’t meet the project’s goals. (0) |
| \* Teams don't have to show the items mentioned above separately. They should be demonstrated by the execution of the functionality of your system  \*\* If it’s not possible to have sections/ features fully functional, there has to be a clear design on how everything will be implemented/ integrated later on | | | |
| Testing (15) | The team shows User Story testing and functional Unit Testing. Team clearly shows the user story testing test plans along with the test results. Also, test scripts (using automated unit testing frameworks or non-automated test scripts) are presented (15) | The team shows User Story testing or functional Unit Testing. Team clearly shows the user story testing test plans along with the test results; or test scripts (using automated unit testing frameworks or non-automated test scripts) (7) | The team shows User Story testing or functional Unit Testing, but the do not present a systematic approach for their testing (3) | The team does not address testing. (0) |
| Agile (5) | The team clearly understands and reflects on areas that the team can improve / has improved. The team attempts to use metrics to demonstrate and compare improvement. The team shows clear action plans to improve on Agile process. | | | |
|  | All three of the above. (5) | Two of the above. (3) | One of the above. (1) | None. (0) |
| Usability Test Plan (10) | The team correctly shows their usability test plan (questionnaire, task instruction for the usability session and evaluation techniques) (10) | The team shows their usability test plan (questionnaire, task instruction for the usability session and evaluation techniques), but the plan is not very clear. (5) | The team partially shows their usability test plan (questionnaire, task instruction for the usability session and evaluation techniques), and the plan is not very clear. (3) | The team does no show their usability test plan (questionnaire, task instruction for the usability session and evaluation techniques) correctly (0). |
| Demonstration (5) | The sections of the demonstration followed logically from one to another. Changes between presenters are done in an efficient and appropriate manner. All team members present. The presentation appeared well rehearsed without any technical issues. (5) | The sections of the demonstration followed logically from one to another most of the time. Most of the time, changes between presenters are done in an efficient and appropriate manner. Most team members present. (3) | The sections of the demonstration followed logically from one to another some of the time. There were some unnecessary changes between presenters and minor technical issues. Only few team members present. (2) | There were constant technical issues. The presentation kept stopping to fix problems. The presentation was disjointed. Only few team members present. (0) |
| Conclusion (3) | The team explains how this presentation will feed into the development process and what the next presentation will be about in terms of the project. (3) | A minor mention is made to future development *and* the next presentation. (2) | A minor mention is made to future development *or* the next presentation. (1) | No mention is made to future development or the next presentation. (0) |
| Question Time + Feedback (2) | The team is able to concisely and clearly answer all of the questions asked by the markers and client.  The team takes note of all of the feedback given. When appropriate, they ask for clarification. (2) | The team is able to clearly answer most of the questions asked by the markers and client. The team takes note of all of the feedback given. (1.5) | The team is able to answer some of the questions asked by the markers and client. The team does not take note of the feedback given. (1) | The team gives incomprehensible answers to questions asked by the client and markers. The team does not take note of the feedback given. (0) |

### 6.5 Final Presentation Assessment

PRB – Week 13 Team Letter: Date: /10/2019 Marker:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Excellent | Good | Satisfactory | Unsatisfactory |
| Punctuality (5) | Some team members arrive early to setup the room and the rest arrive on time. (5) | All team members arrive on time. (2.5) |  | Some team members are late or absent without an apology. (0) |
| Presentation Purpose (5) | The team clearly states why they are there today. What they hope to get out of the presentation and what they’re going to cover in this presentation.(5) | The team clearly states the purpose of the presentation and lists the material that will be covered. (3) | The team lists some of the material they will cover. (1.5) | The team makes no mention of the purpose of the presentation or the material that will be covered. (0) |
| System Overview (5) | The team correctly and accurately identifies all of the benefits the final product provides to the end users. They also identify any client defined constraints they have had to work within. (5) | The team correctly lists most of the benefits the current iteration of the system provides, and any client stipulated constraints that they must work within. (3) | The team correctly lists most of the main benefits the current iteration of the system provides. Some of the descriptions are vague or incomprehensible and may not match exactly with client's intents. (1.5) | The team incorrectly lists the benefits the current iteration of the system provides or fails to mention anything at all. (0) |
| High-Level System Architecture (5)  **Structural View (5)** | The team completely and clearly describes the major hardware components, operating systems, software frameworks, software applications, users and how they relate to each other |  |  |  |
|  | The team clearly shows either |  |  |  |
| * A domain model showing all of the relevant entities and relationships that make up the problem domain. This can take the form of a UML class diagram or an ER diagram; OR | * A UML component diagram that shows the major software components within their software system and the interfaces through which they interact with one another. |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Excellent | Good | Satisfactory | Unsatisfactory |
| System Implementation (50) | The presentation communicates a complete description of the outwardly visible part of the system. The team shows, in detail, that:   * Business goals are met and functionality is complete and consistent with the client’s priorities (25) * Data management is complete in implementation and sufficient data is populated to show system scalability (10) * User interface is in step with current standards, is a professional design and is easy to navigate (10) * The system has been tested. Show the testing methodology used for the system, the coverage of the tests and the results of the tests (5) | The sprint implementation clearly communicates a partial description of the outwardly visible part of the system. The team shows how some of the use stories can be completed with the wireframes. (35)  The team does some testing and addresses some of the following: testing methodology, test coverage and test results. (2.5) | The prototype communicates a partial description of the outwardly visible part of the system. The team indicates how some of the use stories might be completed with the wireframes. (20) | The team does not appear to satisfy any of the project's goals. |
| Agile & Team Maturity (10) | The team clearly understands and reflects on the improvements they have made throughout the year with regards to agile methodologies, team cohesion, technical skills, communication and overall growth. The team uses metrics to demonstrate and compare improvement and evaluates any action plans they have implemented in the past. (10) | The team identifies and reflects on some areas that the team can improve. The team attempts to demonstrate and compare improvement. The team put some plans to improve on Agile process. (7) | The team identifies the need for Agile process improvement. The team understand the need to put plans to improve on their process. (3) | No Agile process improvement. |
| Presentation (10) | The topics of the presentation followed logically from one to another. Changes between presenters are done in an efficient and appropriate manner. The presentation appeared well rehearsed without any technical issues. (10) | The topics of the presentation followed logically from one to another most of the time. Most of the time, changes between presenters are done in an efficient and appropriate manner. (7) | The topics of the presentation followed logically from one to another some of the time. There were some unnecessary changes between presenters and minor technical issues. (3) | There were constant technical issues. The presentation kept stopping to fix problems. The presentation was disjointed. |
| Conclusion Question Time + Feedback (5) | The team explains what they have learned throughout the semester. They summarise what they have done and how they would go on to further improve their system.(2.5)  The team is able to concisely and clearly answer all of the questions asked by the markers and client.  The team takes note of all of the feedback given. When appropriate, they ask for clarification. (5) | The team summarises what they have done throughout the semester. (1)  The team is able to clearly answer most of the questions asked by the markers and client. The team takes note of all of the feedback given. (3) | The team thanks us for our time (.5)  The team is able to answer some of the questions asked by the markers and client. The team takes note of all of the feedback given. (1.5) | The team gives incomprehensible answers to questions asked by the client and markers. |

### 6.6 Source Code Submission Assessment

PRB – Week 13 Team Letter: Date: 31 /10/2019 Marker:

|  |  |
| --- | --- |
| Assessment Criteria | Comments / Marks |
| Source Code Submission:  Submission structured properly and is easy to follow (5)  Implementation follows design. (5) | ( /10) |
| Standards  Implementation Follows the following coding standards:  Formatting  Code Structure  Headers (including authors and revisions)  Function descriptions  Clarity of Input and Output  Object data definition (naming conventions) | ( /18) |
| Consistency  Is everything in the standards section applied across all the source code? | ( /10) |
| Comments  Comment on each module/object up front  Comment on internal source code  Comments distinguished by indentation  Source code indented into logical blocks using left margin and blank lines | ( /20) |
| Reuse  Good usability is evident in function call, data definition, and business rules | ( /10) |
| Installation  Installation procedure/script provided and works  Installation is easy  Directory paths, environment variables not hard coded | ( /12) |
| System Operations  All requirements/functions present and works | ( /20) |

Total ( /100)

### 6.7 System Maintenance Document Submission Assessment

PRB – Week 13 Team Letter: Date: 31/10/2019 Marker:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Assessment Criteria | | | | | | | | | | | | | | Comments / Marks | | | |
| General Presentation:  Documents Titled and Pages Properly Linked/Numbered  Project & Team Identification  Table of Contents  Index & Glossary | | | | | | | | | | | | | | ( /2) | | | |
| Scope  Major Software Functions  Major Design Constraints, Limitations | | | | | | | | | | | | | | ( /3) | | | |
| References  Existing document, System document  Vendor document | | | | | | | | | | | | | | ( /3) | | | |
| Object Oriented Design (32) |  | | | Clarity | | Correctness | | | Standards | | Detail | | Comments | | | |
| High Level Architecture Design (4) | | | (1) | | (1) | | | (1) | | (1) | |  | | | |
| Component / Container / Class structure and their communication (8) | | | (2) | | (2) | | | (2) | | (2) | |
| ERD Diagram and data access strategy (8) | | | (2) | | (2) | | | (2) | | (2) | |
| Sequence Diagrams for implemented User Stories (8) | | | (2) | | (2) | | | (2) | | (2) | |
| Methods Reference (4) | | | (1) | | (1) | | | (1) | | (1) | |
| Each diagram is evaluated by four criteria: correctness, clarity of expression, compliance with standards, and detail. Correctness refers to whether there are any logical errors and inconsistencies in the diagrams presented. Are incorrect relationships being used? Does the diagram accurately represent information? Clarity of expressions refers to how well the diagram is communicated. Are the fonts legible? Are the diagrams cluttered? Are the explanations coherent? Compliance with standards refers to whether or not a diagram complies with standard notations. Is the diagram valid UML? Are the appropriate connector types used? Is the diagram’s notation made up? Detail refers to how fully the view of the system is represented.  Each of these dimensions will be marked as either good, satisfactory, or unsatisfactory which correspond to two, one, and zero marks. | | | | | | | | | | | | | | | |
| User Stories (44) |  | | | | Clarity | | | Correctness | | Standards | | Detail | | | | Comments | |
| User Story Dictionary (4) | | | | (1) | | | (1) | | (1) | | (1) | | | |  | |
| Prototype screens for entire system and Responsiveness design (16) | | | | (4) | | | (4) | | (4) | | (4) | | | |
| User Story Definition (16) | | | | (4) | | | (4) | | (4) | | (4) | | | |
| Flow of Interaction (8) | | | | (2) | | | (2) | | (2) | | (2) | | | |
| Each User Story is evaluated by four criteria: correctness, clarity of expression, compliance with standards, and detail. Correctness refers to whether all is defined precisely and correctly. Standard denotes user story definition; diagrams are following defined standard and structure. Your definition must be detailed enough. As an example, have you covered all acceptance criteria or all aspects in your prototype? Each of these dimensions will be marked as either good, satisfactory, or unsatisfactory. | | | | | | | | | | | | | | | | |
|  | | | Testing Method | | | | Effectiveness | | | | | | | | Coverage | | |
| Software Release Report (15) | | **Usability Test Report (5)** | The tests are precisely defined and the aim and method of the test are appropriately defined  Must include the audience description, metrics definition, approach/plan (might include survey with questions to ask, should include observations of users performing tasks with UI) (2) | | | | Usability Test Report clearly shows how your test has improved your product in terms of usability. A good before-after comparison should be included. (2) | | | | | | | | There are enough observations for all user interface components and their associated functionalities (1) | | |
| **User Story Testing for implemented user stories (5)** | The testing clearly defines the steps required for the user stories, the inputs for each user story, the expected results and pass/fail criteria are clearly and precisely defined. The intended testing procedure is appropriate and clearly defined. (2) | | | | The test cases show that the system operates correctly through the main flow and appropriately handles alternate flows. (1) | | | | | | | | There are test cases for all operations and states for all implemented user story.  (2) | | |
| **Unit Testing (5)** | An appropriate unit testing framework has been selected. Unit tests are clearly defined for each relevant function. (2) | | | | The test cases show that the system operates correctly through the main flow and appropriately handles alternate flows. (1) | | | | | | | | There are test cases for all operations and states for all implemented and relevant functions. (2) | | |
| Special Notes (optional)  Appendix | | | | | | | | | | | | | | | ( /1) | | |

### 6.8 User’s Guide Submission Assessment

PRB – Week 13 Team Letter: Date: 31/10/2019 Marker:

|  |  |
| --- | --- |
| Assessment Criteria | Comments / Marks |
| General Presentation  Document Titled and Pages Numbered  Front page and identification  Table of Contents  Index & Glossary | ( /4) |
| General Description  Title Page  Copyright  Warranties and contractual obligations  Table of Contents  List of illustrations | ( /5) |
| Introduction  Audience description  Applicability statement  Purpose statement  Document usage description  Conventions | ( /5) |
| Introductory Kit  Summary of system features (5)  Getting Started Guide (15) | ( /20) |
| User Manual  Identification of services/functionalities (use case) (5)  For each service/function:  Description  Menus and commands  Screens, navigation, input data, and commands,  Output screens and reports, error handling | ( /44) |
| Installation Guide | ( /10) |
| System Administrator Guide | ( /10) |
| Appendices, Bibliography, Glossary, Index | ( /2) |

Total ( /100)

## References

1. Schwalbe, K., Information Technology Project Management, 4th ed. 2006, Thomson

2. Stellman, A. and Greene, J., Applied Software Project Management. 2006, O’Reilly

3. Sommerville, I., Software Engineering, 9th ed. 2010, Addison Wesley.

4. Pressman, R., Software Engineering: A Practitioner's Approach. 6th ed. 2005, McGraw Hill.

5. Quatrani, T., Visual modeling with Rational Rose 2000 and UML. Object Technology, 3rd ed., 2002, Addison Wesley.

6. Oestereich, B., Developing Software with UML. 2nd ed. 2001, Addison Wesley.

7. Schach, S., Object-Oriented and Classical Software Engineering. Fifth ed. 2002: McGraw Hill. 628.

8. Bennet, S., J. Skelton, and K. Lunn, UML. Schaum's Outlines. 2001: McGraw Hill. 360.

9. Fowler, M. and K. Scott, UML Distilled. Object Technology. 2000: Addison Wesley. 185.

10. Pooley, R. and P. Stevens, Using UML Software Engineering with Objects and Components. Object Technology. 1999: Addison Wesley. 256.

11. Nakhimovsky, A. and T. Myers, Professional Java XML Programming with Servelets and JSP. 2000: Wrox. 772.

12. Cagle, D., et al., Beginning XML. 2001: Wrox Press Ltd. 785.

13. Cauldwell, P., et al., Professional XML Web Services. 2001: Wrox Press Ltd. 803.

14. Chang, E., T. Torabi, and W. Rahayu. Issues and Solutions in Running a Full Year Software Engineering Project for Computing Majors. in Software Engineering Education and Practice, SEEP 98. 1998. Dunedin, New Zealand.

15. Torabi, T. and G. Claxton. Developing an Applied Computing Course for Business Students. in NCCQA Conference. 1999. Auckland, New Zealand.

16. Nakamura, Y., et al., XML and Java: Developing Web Applications, Second Edition. 2002: Addison Wesley. 661.

1. : Statement of Authorship

****

**Department of Computer Science and Information Technology**

STATEMENT OF AUTHORSHIP FOR GROUP ASSIGNMENTS

Please refer to LMS for a downloadable copy of the “Group Statement of Authorship”. All students must fill in and sign this statement. The statement must be included with each group submission.

1. : Statement of Effort



**Department of Computer Science and Information Technology**

**SAMPLE STATEMENT OF EFFORT FOR GROUP ASSIGNMENTS**

|  |  |  |
| --- | --- | --- |
| **Section** | **Page** | **Members contributed** |
| 1.0 Scope |  | John |
| 1.1 Major Software Functions |  | Jack |
| 1.2 Major Design Constraints and other Requirements |  | George |
| 2.0 Reference Documents |  | Val |
| 2.1 Existing Software Documentation |  | John |
| 2.2 System Documentation |  |  |
| 2.3 Vendor Documentation |  |  |
| 3.1.1 Design of User Management Component |  | Jill |
| 3.1.1 Design of Reference Management Comp |  | Jack |
| Etc | etc | Etc |
| 5.0 Object Dictionary |  | Val, George |
| Etc | etc | Etc |
| Chief Designer and proof reader |  | Jack |
| Printing, binding, … |  | Joint |
| Etc | etc | Etc |

|  |  |  |  |
| --- | --- | --- | --- |
| **Student Name** | **Student Id** | **Signature\*** | **Date** |
| John Smith | 0121212544 |  |  |
| Jack | 0547185852 |  |  |
| Val | 5870052688 |  |  |
| Etc | Etc |  |  |

\* MUST be signed by all members

1. : Document Change Record

The following table records the complete history of the successive editions of the present document.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Ver. | Edition Date | Reason for change | Pages affected | Edited by |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

1. : Guidelines and Procedures for Major Submissions

### Introduction

The purpose of this document is to set a standard procedure and structure for the documentation and software submitted by the Industry Project teams. This would provide the better evaluation, management, and archive for the project software and documents.

**Note:** Each document must include a summary "*statement of effort*" outlining who has contributed to each section or task for preparation of the document.

### Where to Submit

By the due date, each team has to submit:

1. An *electronic submission*, as it is outlined in lectures and sprint meetings.

### Electronic Submission

Directory Structure for Electronic Submission:

It is suggested that from the very beginning, each team organise all their files in directories that will be easy for team members to manage. For example all final copies of files for submission can be structured in a "submission" directory.

**Note:** For the purpose of easily managing your files for submission and future use and references to these documents you may start building such directory structure as sample (refer at the end). The final copy of documentation and code produced by the team for submission can be placed in the appropriate sub-directory.

1. : Directory Structure for Final Submission

For the purpose of **easily managing** your submission and **future use and references** to these documents, it is recommended that the submission directory -say PROJECTnn- to be structured as shown below. The documentation and code produced by the team should be placed in this directory structure as suggested in here.

Please note that the content of directories may slightly change based on the type of the project. If any question please consult your supervisor for help.

### Suggested Submission Directory Structure:

###### Root Folder:

README.txt: Outlining all submissions made + any additional information regarding the submission(s).

###### Directory: Source

README.txt: Outline how to load, compile and run your system. (Including how to setup any db alias and configuration(s))

All project related files with your database files (or full db script to create the db with some sample data) .

Include all under your web server + any other servers. Eg: If you're using Apache Web server, include all directories under Apache (conf, cgi-bin, htdocs etc) under the source directory.

Sub directory: Client

All HTML, PHP/JSP, additional script files, etc.

Sub directory: Server

All HTML, PHP/JSP, additional script files, etc.

###### Directory: Documents

Sub directory: Submission\_1

User Document

System Maintenance Document

Testing Documentation (can be submitted separately)

Sub directory: Submission\_2

Revised & Updated User Document

Revised & Updated System Maintenance Document

Revised & Updated Testing Documentation (can be submitted separately)

###### Directory: Database

* dbREADME.txt
* All db files (including scripts/data dump etc) with sample data set. Include DDL scripts to install a starter dB on the server side (with some initial data (eg: default admin user)).

###### Directory: Setup

* Information on your directory structures, compilation, installation, any make file, or script.   
  Include one-or-more of the following so that your client can configure the software easily.
* Setup instructions (server and (any) client side)
* Installed Shield Express (For NT installation)
* Script files (Eg. make file)
* Any plug-ins needed (and instruction on how to install them)
* Automated script files to install major components etc.

###### Directory: Progress

Sub directory: Unit\_A

Meeting Agendas (ordered by week)

Meeting Minutes (ordered by week)

Team Weekly Log reports (ordered by week)

Individual Weekly Progress reports (ordered by week & by team member)

Sub directory: Unit\_B

Meeting Agendas (ordered by week)

Meeting Minutes (ordered by week)

Team Weekly Log reports (ordered by week)

Individual Weekly Progress reports (ordered by week & by team member)

###### Directory: misc

Any other additional documents, documentation, vendor documents, etc. which you think are important.