

# Software Engineering Components for CS3 Capstone Projects 2019

## 1. Introduction

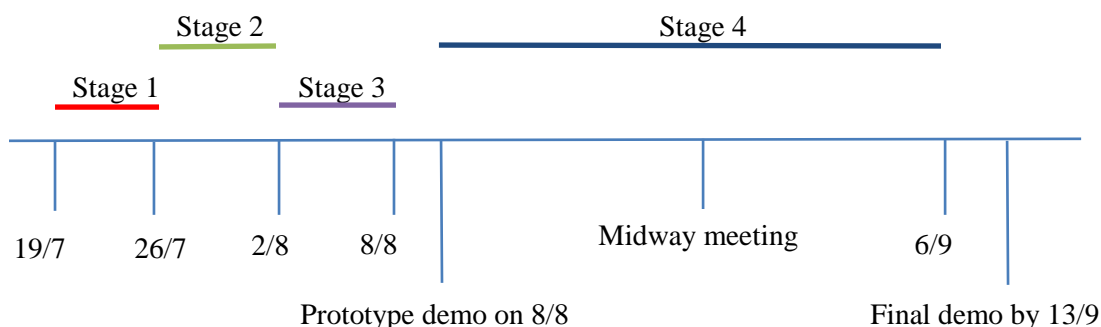
In the second semester of their third year all students majoring in computer science undertake a *three-* person capstone project focussing on delivering a working prototype system designed and implemented according to established software engineering techniques in object-oriented programming. Students doing the games course (CSC3020H) together with CSC3003S *in the current year* will add an explicit software engineering aspect to their games projects.

Games students choose their project in the first semester, all other students are given a list of possible projects towards the end of the first semester. At the start of the second semester students who are not doing a games project concurrently provide their top five project choices and indicate who their project team is.

CSC3020H project groups each tackle a different project. CSC3003S project groups working under the same supervisor (playing the role of client) tackle the same project. *All students* have to produce the software engineering deliverables listed here.

## 2. About the Projects

1. The programming language can be determined by the client/supervisor. *All groups doing a particular project have to use the same object-oriented language.*
2. The capstone project is about software engineering (not research). The project forms the high point and final application of the students' undergraduate learning of computer science and software engineering.
3. The project has to be demonstrated during development and on completion. Should a project not be able to run in the senior laboratory then it is the student group's responsibility to bring the required machine to the department for the demonstration. Failure to run for whatever reason will be equivalent to failure to demonstrate.
4. Several intermediate handins on the design process are required.
5. A project report is required. A document describing what is necessary in this report is available separately.
6. Students are as always expected to complete a non-plagiarism declaration. Students have to write their own code and report, or make suitable references for small parts of either from another source.
7. Time:
  - Students are expected to spend 10 hours per week on the project.
  - Students doing CSC3020H spend an additional 5 hours a week on their project.



### 3. Stage 1: *19<sup>th</sup> July – 26<sup>th</sup> July*: Project Start up

#### 3.1. Objective

- To establish and understand agreed framework, objectives, schedule and scope for the project

#### 3.2. Tasks

- Finalize goals and prepare statement of scope.
- Finalize roles of the team members
- Agree meeting schedule with client (first three meetings are fixed: Friday 19/7, 26/7 & 2/8, with a demonstration on 8/8).
- Start and present project plan for 6–7 week (+ 1 week vac) project to client. The plan will have three iterations showing the spread of use-cases across iterations and the allocation of SE tasks (analysis, design, coding, testing).
- Initial agreement on development environment (including programming language) with client and document outcome.
- Start to prepare use cases, analysis model and design model
- Risk analysis: identify areas of risk and mitigations.

#### 3.3. Output — numbered items (in bold) are to be handed in, see marksheets for details

1. **Statement of Goals (Scope)**
2. **Statement of Roles**
3. **Risk report**

- First draft of Project Plan
- First drafts of use cases and analysis model and object interaction (dynamic models).

#### 3.4. Duration

10 hours

#### 3.5. Weighting: 10% of project mark

The project is about 50% of the practical work of CSC3003S (see course handout).

**Hand-in Date:** 9h00 Friday 26<sup>th</sup> July

### 4. Stage 2: *26<sup>th</sup> July – 2<sup>nd</sup> August*: Planning and Modelling

#### 4.1. Objective

- Complete the initial design of the system.

#### 4.2. Tasks

- Create Use-Case Scenarios, each consisting of a use-case diagram in UML notation and an associated narrative. Your scenarios should be structured hierarchically with a high-level use-case subdivided into one or more sub-cases.
- Create an analysis model
- Develop object interaction from use cases
- Complete the project plan
- Prepare weekly progress report
- Decide what kind of prototype is to be implemented: throw-away, evolutionary or agile.
- Create a test plan.
- Start prototype design and implementation.

#### 4.3. Note

- You must use software engineering tools such as Microsoft Visio (installed in the senior lab), [ArgoUML](#) and [GanttProject](#) to create all artefacts. The department also has copies of Microsoft [Project](#) and [Visio](#) available for students under our MSDN agreement. These run only

on windows. Please complete the agreement form (obtainable at our administrative office) and then collect the software from Sam Chetty.

- Hand written documents will not be marked.

#### **4.4. Output — numbered items (in bold) are to be handed in, see marksheets**

- Use case diagrams
- 1. **Use case narratives**
- 2. **Analysis model**
- 3. **Dynamic model (interaction or sequence diagram or state machine)**
- 4. **Project Plan**
- 5. **Test plan**
- Progress Report (handed in at the next stage)

#### **4.5. Duration**

10 hours.

#### **4.6. Weighting: 10% of project mark**

**Hand-in Date:** 9h00 Friday 2<sup>nd</sup> August

### **5. Stage 3: 2<sup>nd</sup> August – 8<sup>th</sup> August: Prototype**

#### **5.1. Objective**

- To create prototype of product and demonstrate to client.
- Revise and update documentation.
- Prepare weekly progress report

#### **5.2. Tasks**

- Using use case artefacts and initial design, create prototype for demonstration to client
- Document the decision on choice of prototype

#### **5.3. Output (refer to marksheets)**

1. **Progress Report (submit this stage and the previous stage progress report).**
2. **Statement and justification of choice of prototype.**
3. **Prototype source code.**
4. **Demonstration to client to see to how the team is progressing with the client requirements in terms of scope. The client says if this is in the right direction and may perhaps suggest modifications.**
5. **Schedule further meetings with client including a midway meeting in Stage 4 to demo progress and discuss the final report.**

#### **5.4. Duration**

10 hours.

#### **5.5. Weighting: 14% of project mark**

**Hand-in Date:** 9h00 Thursday 8<sup>th</sup> August

**Demonstration Date:** Thursday 8<sup>th</sup> August

### **6. Stage 4: 12<sup>th</sup> August – 6<sup>th</sup> September: Implementation and Testing**

#### **6.1. Objective**

- Implement all use cases and test product
- Document Project

#### **6.2. Tasks**

- Complete implementation phase.

- Carry out test plan. Record results for all test cases.
- Arrange meetings with client and tutor. It is recommended that you **meet midway** through this period (early September) for a demonstration of progress and discussion of the requirements for the final report.
- Write final project report.
- Write user manual.
- Regular team meetings and at least fortnightly meetings with tutors.
- At least one further meeting/demo with the client to follow, before the final handin.

### 6.3. Output (refer to final report guidelines)

- **Test results**
- **Final product**
- **Final Report & User Manual**

### 6.4. Duration

27 hours

## 7. Complete Project

Submit your **final project report** according to the requirements that will be provided.

*Weighting:* **66%** of project mark (about  $\frac{1}{3}$  of the practical mark for the course)

**Hand-in Date for Report and Software:** 9h00 Friday 6<sup>th</sup> September

**Demonstration Date for Software:** Between 9 and 13 September.

- Software Engineering counts half of the final project mark. The software itself counts for the remaining half.

## 8. Summary of Marking Scheme:

<i>Artefact</i>	<i>Marks</i>	<i>Due (9h00)</i>
1. Project Scope, roles & risks (§3)	10%	26 <sup>th</sup> July
2. Project Planning & Modelling (§4)	10%	2 <sup>nd</sup> August
3. Prototype & progress reports (§5)	14%	8 <sup>th</sup> August ; Demo same day
4. Final Report (§6 & §7)	66%	6 <sup>th</sup> September; Demos: 9 <sup>th</sup> to 13 <sup>th</sup> September

## 9. Note:

- Meetings with clients/supervisors will be scheduled on Fridays for the first three weeks; subsequent meetings will be scheduled by clients depending on their schedules. The tutor assigned to your project will assist in arrangements for this.
- The prototype demo is scheduled by default for 8<sup>th</sup> Aug at 9AM, but other times may be arranged on that day.
- The times for final demos depend on the client's schedule but have to be completed before the latest dates indicated above. The software demonstrated must, however, be the same as that submitted.
- The prototype and final product must be submitted on Vula by the dates shown, but the demos will take place at times that are convenient for the client.