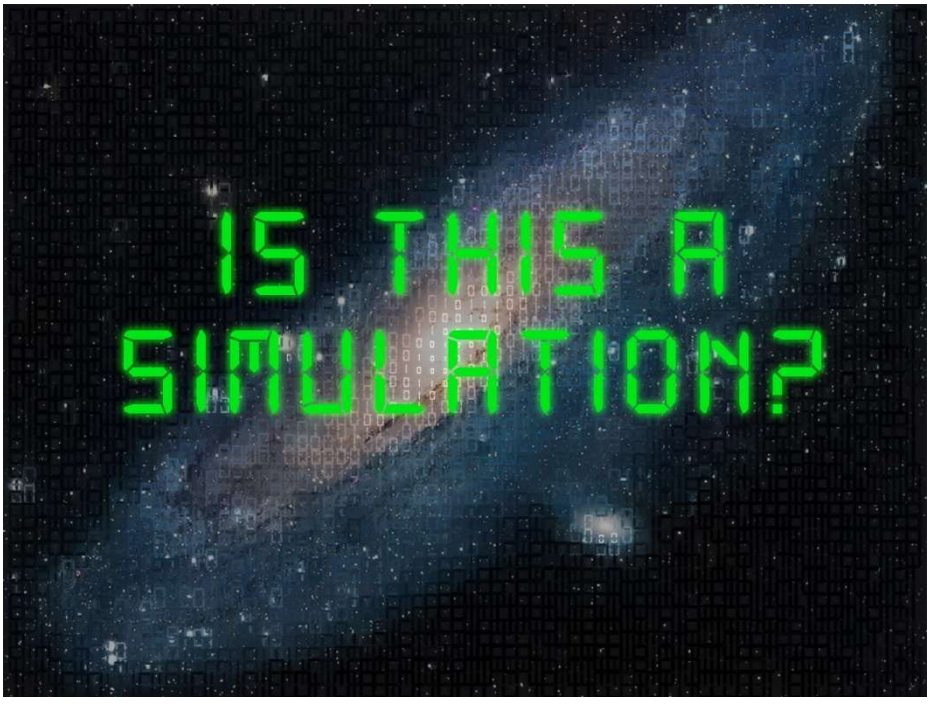
ProCP

A simulation application

Workbook



September 2020

Preface

In this document you will find a description of the ProCP project.
You will find the assignment, week schedule and other project requirements and information.

To be able to work successfully in a project it is very important that you can work together with fellow-developers and that your level of communication is sufficient. For this reason it is important that you are present during all lessons and project meetings.

We hope you will enjoy this module.

Table of contents

1 General plan

1.1 *Objectives*

After successfully finishing this project you will be able to:

- successfully collaborate with team members to deliver a single non-trivial application,
- take ownership of the project by being proactive and in-control,
- make a realistic planning and honour the promises made with peers and tutor/client,
- apply (learned) knowledge to make justifiable decisions for both technical and process related challenges,
- make use of feedback from peers and tutor/client to further improve as a professional,
- make use of artefacts and tools to support an iterative way of working.

1.2 Required knowledge

To be able to attend this module, the student:

- Must have passed the propaedeutic phase;
- Must have at least have a grade for OOD1;
    - You will be default 'marked' as conditionally participation when your OOD1 grade is lower than 4.5

When in doubt, the teacher can decide whether a student can **conditionally** participate in this module. This means that you will be removed from the project when you are not able to equally contribute to the project compared to the rest of your team members.

1.3 Organization

The project is carried out in a project team consisting of 6 students.
Presence at the classes and meetings is mandatory. Missing 2 or more times during the entire semester will result into a grade of 1 and removal from the project.

Every week there are some lessons for ProCP on the schedule.
Every week in one of these lessons there is a meeting with your group and the tutor to talk about your project. It is expected all these group-meetings will be during the scheduled lesson-time.
The rest of the scheduled lessons should be used to work on the project.

In case many groups are interested in information about certain subjects, we will schedule a lesson for all groups at the same time. These lessons will take place during the hours scheduled for ProCP.

1.4 Grading

The grading of ProCP is based on the following criteria:

1. Approval of project proposal & plan;
2. Active and involved participation;
3. Presence during classes and project meetings;
4. Quality (sub-)deliverables.

If your group failed criteria 1 your work for ProCP is finished and the final mark will be 1.
For criteria 2 & 3: If you are not present for the first time: you will get a warning. If you are not present for the second time: your work for ProCP is finished and your final mark is 1.

If you pass criteria 1, 2 and 3 then and only then your mark for the project will be determined by:

	Weight
Kick-off phase: - Project plan.	Go or No-Go
Initial phase: - URS; - Work division report; - Plan for iteration 1.	10%
Iteration 1: - Updated URS; - Working application(s) + source code; - Updated work division report; - Plan for iteration 2.	20%
Iteration 2: - Updated URS; - Updated design document; - Updated working application(s) + source code + unit tests; - Updated work division report; - Plan for iteration 3.	25%
Iteration 3: - Updated URS; - Updated design document; - Updated test report; - Updated working application(s) + source code + unit tests;	35%
End phase: - Process report and presentation	10%

The final mark will be rounded to the nearest multiple of 0.5. For this module there is no final exam.

2 The assignment

SIM Software Inc. is a fast growing company aiming at innovative solutions for simulation problems. In the last few years, SIM software has focused on traffic simulation software, but the company would like to extend its expertise to cover a broader area of simulation software. So the company asks for other project proposals in the area of simulation software.

You as a group of young innovative software professionals are requested to come up with a proposal for an application in the area of simulation software. SIM software is open for different types of applications, however, they all must adhere to the following requirements:

- The User Requirements Specification (URS) for the application shall contain both functional and non-functional requirements.
- The Design Document for the application shall contain UML class diagram(s), descriptions and UML sequence diagram(s).
- The application should perform certain simulations: i.e. it should implement a simplified model of some objects in the real world, and contain some randomization to simulate possible external events that influence the state of the modeled objects.
- It should be possible to store simulation models and results in a file or database, and load previously stored models and results from that file or database.
- In the design of the application, the most important OO concepts are needed, like classes, objects, properties, interfaces, inheritance and events.
- The application will be implemented in an object-oriented language like C# or Java.

SIM Software is interested in following simulation applications:

- *City traffic simulation*: A city planner can configure the roads and crossings in a city to simulate traffic and pedestrian flow during rush hour.
  The simulation provides the means to optimize the configuration of roads, crossings and traffic lights via statistics related to how the traffic resolves.
- *Fire escape simulation*: A floor designer can configure a floor with the amount of people and fire extinguishers to simulate an event when fire(s) occurs and spreads through the floor.
  The simulation provides the means to optimize the placement of fire extinguishers via statistics related to how many the people reacted in the simulation.
- *Airport luggage simulation*: A manager can configure the resources (employees, carts, etc.) required to simulate the processing of luggage from arriving and leaving airplanes.
  The simulation provides the means to determine the optimal resource allocation via statistics related to how the luggage has been processed.
- *Shop distribution simulation*: A franchise owner can configure the placement of shops and warehouses to simulate the required routes and time to (re)stock

the products.
The simulation provides the means to determine the optimal placement and distribution on a map via statistic related how the demand of customers has been handled.

- *Town hall simulation*: A manger can configure the counters and queues in a town hall to simulate the process of handling different resident appointments/requests.
  The simulation provides the means to determine the optimal division of counters via statistics related to how fast the queues have been handled.
- *Mini-mall simulation*: A CEO can configure the shops in a mini mall to determine the profit of each shop.
  The simulation provides the means to determine if the placement of shop is optimal via statistics related the income of each shop.

You can decide which application is interesting for you. After you made your selection you will have to specify the functionalities and present your proposal to your tutor, who represents the board of management from SIM Software Inc. Your tutor will decide whether your proposal will be accepted as is, requires some modifications, or is rejected.

In order to be able to start working on your project, your project proposal should first be accepted. Of course, the more innovative your proposal is, the higher it may be rewarded in the end.

3 Week schedule

Each week it is expected that you work on the project and have a mandatory meeting. It is expected that you work on both the documentation as the implementation.

Work on project
Note that the project actually exists out of 7 phases (kick-off, initial, iteration 1, calibration session, iterations 2 &3 and end), which are finalized with an assessment of the required deliverables and group work.

To structure the work, we have specified activities and deliverables. The first period has a suggestion for activities per week, while the second period we expect you to come up with a schedule.

Take note that for **deliverables** marked in **bold** and yellow, it is mandatory to hand it in before that week's deadline; this can be done via email or a link to the tagged commit in GIT. **The deadline for marked deliverables is the Monday of the next week, before 08.00 AM**. For example: if the deliverable is marked in week 2, the deadline is the Monday of week 3 before 8.00AM.
Missing a deadline results into 0 points for the deliverables/phase.

Note that the unmarked deliverables are a suggestion. When you decide to hand it in, you can ask feedback about that deliverable.

Feedback for deliverables
This project is about you showing your proficiency as a software engineer. For this reason, during each phase, you can only ask for feedback about a deliverable **once** (e.g. during iteration 1: once about the URS, once about the design document, etc. During iteration 2: once about the URS, once about the design document, etc.).

Be **specific** about which part you want feedback (e.g. feedback for use case 1 about the MSC). When you do not specify this, the teacher does not know what is expected and cannot prepare for this and has to do this during the meeting.

Meetings
For all the project meetings you are expected to be present and make a meeting agenda and (afterwards) meeting minutes. A meeting should not take more than 30 minutes. Absence will have the same consequences as with ProP.

The meeting agenda should be distributed via e-mail to all persons invited for that meeting (i.e. also to your ProCP tutor). Of course, any documentation you would like to discuss during the meeting should be distributed together with the meeting agenda. The deadline for sending the agenda is **one working day before the meeting, before 08.00 AM**.

During every meeting you have to make meeting minutes, which should be distributed via e-mail to all participants of that meeting, **latest one working day after the**

meeting took place.

Finalizing a phase
You will get one subgrade, after successfully finishing each phase (excluding kick-off phase). Any changes to the deliverables after a deadline will not affect the already given subgrade for those deliverables.

It is expected you continue updating the deliverables as you progress through the project, but the improvement will only affect the subgrade of the phase you are in.

4 ProCP schedule

This chapter shows what the deadline of each marked **deliverable** is. The activities and unmarked deliverables are a schedule suggestion for your group.

4.1 First period

Kick-off phase

Week 1

Activities: • Startup / Form groups • Come up with a proposal for an application • Research the to-be implemented algorithm/application • Create draft of project plan
Deliverables: • Concept version of project plan • **Proposal for an application**

Week 2

Activities: • Research the to-be implemented algorithm/application • Update concept version of project plan • Create concept version of URS
Deliverables: • Concept version of URS • **Final version of project plan**

Initial phase

Week 3

Activities: • Discuss project plan • Create concept version of plan for iteration 1 • Updated version of project plan & URS
Deliverables: • Updated version of URS • Concept version of plan for iteration 1

Week 4

Activities: • Discuss URS & plan for iteration 1 • Updated URS & plan for iteration 1 • Create work division report
Deliverables: • **Final version of plan for iteration 1**

• **Updated version of URS** • **Work division report**

Iteration 1

Week 5

Activities: • Present URS to you tutor • Discuss • Update URS
Deliverables: • Updated version of URS

Week 6

Activities: • Code • Create concept version of plan for iteration 2
Deliverables: • Concept version of plan for iteration 2

Week 7

Activities: • Code • Debug • Update work division report
Deliverables: • **Final URS for iteration 1** • **Final version of plan for iteration 2** • **Source code of proof of concept** • **Proof of concept** • **Updated version of work division report**

Week 9

Activities: • UML Class diagram(s) & the non-trivial sequence diagram(s)
Deliverables: • **UML Class diagram(s) & non-trivial sequence diagram(s) of proof of concept**

4.2 Second period

Calibration session

Week 10

Activities: • Present proof of concept to board of tutors • Calibration session about how to proceed • Update URS • Create design document
Deliverables: none

Iteration 2

From week 11 to 13

Activities: • Determined by group
Deliverables: • **Final URS & design document** • **Final version of plan for iteration 3** • **Source code of prototype** • **Unit tests of prototype** • **Prototype** • **Updated version of work division report**

Iteration 3

From week 14 to 16

Activities: • Determined by group
Deliverables: • **Final URS & design document** • **Source code of final product** • **Unit tests of final product** • **Final product** • **Final version of work division report** • **Process report (including work division)**

End phase

Week 17 or 18

Activities: • Final presentation followed by a final meeting about the marks.
Deliverables: • Presentation

5 Expectation matrix

During the entire project the teacher will look at the work the project group has done, but also what each individual contributed. Based on the situation and observations a (sub)grade will be compiled and given.

To give more insight into this process and what your group can expect, a global overview will be given. In the overview a task is described and what the expectation(s) are to be eligible for a specific range of (sub)grade.
Note: to be eligible for *Leading (>7),* your group also needs to fulfil everything from the cell in *Developing (5.5 ~ 7).*

It is important to realize that the matrix is indicative and depending on the situation the teacher can deviate from it. Also keep in mind not every task is as important and different weights are given to them.

5.1 Applicable during the entire project

Task	Experimenting (<5.5)	Developing (5.5 ~ 7)	Leading (>7)
Feedback	Did almost nothing with the feedback	Sufficiently applied the given feedback	Critical attitude about the feedback, the meetings and system exceeded the given feedback
Soft skills - Work ethic	The team did not apply the theory covered by the university courses. Team members and hours are present, but it is not clear who did what and when Team members worked in a re-active manner	The team applied the theory of the university course. A weekly overview is given about how much time a team member has spent on a task Team members worked in an active and involved manner	The team took justifiable risks to include new technologies, techniques, tools, etc. to supplement their work. It is easy to determine how the work was divided between team members and if the actual hours spend matches the expected hours. Team members worked in a pro-active manner
Soft skills - Documentation	Documentation does not contain the	Documentation contains the	Documentation contains the right

	required content covered by the theory It is not clear what has changed/updated in the new version The layout of the documentation is unprofessional and lacks elements like front page, TOC , etc.	required content covered by the theory It is clear what the history is of the documentation and what has been changed.	amount of details and is structured a good way.
Soft skills - Meetings	The meeting are badly prepared and chaotic. The minutes cover only parts of the content of the meeting.	The meetings are prepared and structured. The minutes contain the decisions made and feedback supplied during the meeting. Agenda and minutes are distributed on time	The meetings are dynamic with group members involved in the discussions. The minutes also contain details and covers the entire meeting.
Soft skills - Misc.	Project work required initiative from the teacher and the project group needed 'external motivation' to deliver on time/deadlines are missed. A lot of guidance is required for creating the deliverables and it was not clear how the progression was.	Project group followed the student workbook and works together on the project. At least a source control system & IDE are used and at most one deadlines is missed. Some help was required for creating the deliverables, but the group had a clear understanding how to do it. It was clear what the progression was during the iteration.	Project group took initiative and are proactive. New idea are proposed and are not 'afraid' to go outside their comfort zone. Appropriate tools are used to support their workflow and no deadlines are missed. The actual process of creating the deliverables is a blackbox, but when asked about it it is clear the group has given a good

			thought about it. It was transparent how the progression was during the iteration (e.g. using tools to give the transparency).

5.2 Kick-off

Task	No-go	Go
Project plan	The scope of the project does not follow the minimum requirements of ProCP. It is not clear what the project will be about and how it will be run. The project plan is not following the theory of PM.	The scope of the project meets the minimum requirements of ProCP. The project plan is following the theory of PM and it is clear how the project will be run.

5.3 Initial phase

Task	Experimenting (<5.5)	Developing (5.5 ~ 7)	Leading (>7)
URS	The combined use cases does not cover the core of the system and/or is not clear how they. There is a good separation between functional and non-functional requirements.	The combines use cases covers at least the core of the system and are written in a clear way. Each use case is of the correct scope and is complete. Extensions cover prevalent interaction and are handled. The use cases describe the same system/are coherent.	The use cases also functionalities exceeding the core of the system. The steps written in the use cases are complete and of correct depth. GUI designs are present and matches the use cases. It is clear on what level a use case is and the priority given makes sense.
Plan for iteration 1	Not clear why and/or which use case(s) will be implemented during iteration 1.	The selected use case(s) are appropriate for iteration 1 and form a Proof of Concept	An analysis is given why the selected use case(s) are chosen and why they belongs to the Proof of Concept.

5.4 Applicable during iteration 1

Task	Experimenting (<5.5)	Developing (5.5 ~ 7)	Leading (>7)
Proof of concept (PoC)	The PoC does not match the project plan and/or URS	The PoC proves that the core of the system is feasible and matches the documentation.	The PoC also contains extras; If the problem analysis contains additional functionalities these are (first) implemented.
Plan for iteration 2	Not clear why and/or which use case(s) will be implemented during iteration 2. Unfinished functionalities 'disappeared'	The selected use case(s) are appropriate for iteration 2 and combined with the PoC form a Prototype	An analysis is given why the selected use case(s) are chosen and why they belong to the Prototype

5.5 Applicable during iteration 2

Task	Experimenting (<5.5)	Developing (5.5 ~ 7)	Leading (>7)
Design Document - Analysis	Bad OO design for the diagrams and it is clear that it is not possible to create the system based on the design.	The diagrams are updated and improved with new insights and now also include the newly selected uses case(s). The updated class diagrams & sequence diagrams conveys how the implementation will be done in a correct way.	Extra related functionalities are added to the analysis and together they form Prototype Plus. The UML standard is applied in the diagrams and it is possible to create the Prototype based on the diagrams! There is a loose coupling, but high cohesion in the design (for example via design patterns, separation of concerns and packages).
Prototype (Proto)	The Proto does not match the URS, design document or acceptance test.	The Proto has the functionalities putting it in a prototype state. The documentation and system matches.	The Proto also contains extras; If the problem analysis contains additional functionalities these are (first) implemented.
Plan for iteration 3	Not clear why and/or which use case(s) will be implemented during iteration 3. Unfinished functionalities 'disappeared'	The selected use case(s) are appropriate for iteration 3 and combined with the Proto makes the system production ready. When not all use case(s) will be implemented a justification is present.	An analysis is given why the selected use case(s) are chosen and why it belongs to the final product.
Unit tests	Unit tests only covers a part of the	Unit tests covers almost all the	Non-trivial unit tests covers all the

	discussed requirements with business logic. The unit test are 'entwined' resulting into not testing each unit separately. It is not clear how well the simulation algorithm is tested.	discussed requirements and business logic. The unit test tests each unit separately. The core of the simulation algorithm is tested, but it might miss some edge cases tests.	discussed requirements, business logic and edge cases. Additional types of testing is performed, such as acceptance testing.
Bonus	Group was not able to deliver what was promised	Promises are met	The group delivered more than promised and the extras are useful/make sense in the context. New technologies used for the project.

5.6 Applicable during iteration 3

Task	Experimenting (<5.5)	Developing (5.5 ~ 7)	Leading (>7)
Design Document	Bad OO design for the diagrams and it is clear that it is not possible to create the system based on the design.	The design of the system has been refactored and at least one (logical) design pattern is applied. The updated class diagrams & sequence diagrams conveys how the implementation will be done in a correct way.	The UML standard is applied in the diagrams and it is possible to create the final product based on the diagrams! There is a loose coupling, but high cohesion in the design (for example via design patterns, separation of concerns and packages).
Final product	Previously discovered problems & bugs are partly or not solved. The system does not match the URS, design document or acceptance test.	Previously discovered problems & bugs are resolved. The system contains all the agreed functionalities. The documentation and system matches.	The system also contains extras.
Unit tests	Unit tests only covers a part of the discussed requirements with business logic. The unit test are 'entwined' resulting into not testing each unit separately. It is not clear how well the simulation algorithm is tested.	Unit tests covers almost all the discussed requirements and business logic. The unit test tests each unit separately. The core of the simulation algorithm is tested, but it might miss some edge cases tests.	Non-trivial unit tests covers all the discussed requirements, business logic and edge cases. Additional types of testing is performed, such as acceptance testing.
Bonus	Group was not able to deliver what was promised	Promises are met	The group delivered more than promised and the extras are useful/make sense in

			the context. New technologies used for the project.

5.7 End phase

Task	Experimenting (<5.5)	Developing (5.5 ~ 7)	Leading (>7)
Presentation	The scope of the presentation does not cover the system and project sufficiently. The presentation is not at a correct 'level' for the target audience. The length is a lot shorter or longer than 10 minutes	The scope of the presentation at least cover the realized system. A demo is also included. The presentation is at a correct 'level' for the target audience. The length of the presentation is about 10 minutes	The presentation includes extras like understandable algorithm explanations, justification of (big) decisions made, etc.
Process report	The process report does not contain all the required content.	The process report look like a 'professional' report and describes what and how much time was spend per project member per week. Each project member has written a personal reflection.	The process report gives a good overview how the process was and what the work division was. The included personal reflections contain critical thinking. The reflection covers topics like what went good/wrong with the work done, how to improve for future work, etc. A chapter about DOT Framework is included covering which strategies are used for the work done.