Studienarbeit: Reverse Engineering Lab

Studiengang Informatik
OST - Ostschweizer Fachhochschule
Campus Rapperswil Jon

Semester: Autumn 2022

Autors: Gianluca Nenz
                      Ronny Mueller
                      Thomas Kleb

Project Advisor: Ivan Buetler
Release: E-Prints
Version: Monday 17th October, 2022

School of Computer Science
OST Eastern Switzerland University of Applied Sciences

Abstract

Contents

Chapter 1

Project Idea

dasdasd

Chapter 2

Management Summary

Chapter 3

Product Documentation

Chapter 4

Project Documentation

4.1 Project Plan

4.1.1 Project Overview

The goal of this project is to create and organize a lab, which shows and explains future students of the Ostschweizer Fachhochschule (OST) how reverse engineering is performed and which tactics are used to get information out of a program. To accomplish this task, the lab will have several exercises organized in the different domains. These exercises will be accessible through the Hacking-Lab hosted on the OST server.

Hand-In

The finished Report will be handed in according to the rules set by the "Studiengangsleitung Informatik" and the supervisor:

- The PDF version will be sent to the advisor and to the OST archive.

- The printed version will be handed in to the supervisor for reading and grading.

4.1.2 Management

Time Management

The project started on the first week of the semester (KW 38) and ends in week 51 giving us around 14 weeks to be done with the Hand-In.
Since the module has a total ECTS of 8 each of the students has to work around 240h during the semester which can be seen in table 4.1 together with the total planned time investment. This means, that per week each student should work around 17.1 hours.

Name	ECTS	Time spent per Week [h]	Total Time spent [h]
Gianluca Nenz	8	17.1	240
Ronny Mueller	8	17.1	240
Thomas Kleb	8	17.1	240
Total	32	52.3	720

Table 4.1: Time Investments

Planning

In the past modules Software Engineering Practices 1 and 2 (SEP 1 + 2) we were introduced to different ways to plan and organize a project. The main tools we learned, RUP (Rational Unified Process) and Scrum, are mainly used in software development but can be adapted to other projects aswell. They both use different aspects of time management and organisation which is why we intend to apply them to our project.

RUP (Rational Unified Process)

RUP is a process which allows a team to distribute a longterm plan over a given time and section it into multiple steps and parts. RUP has four main phases which have characteristics to be taken care of:

Inception: The goal in this phase is to propose the project, identifying the criterias to be assessed and a first estimation of risk and success. This part should take less than 5% of the total time.
Elaboration: The elaboration phase is used to identify the requirements and the architecture. It should also be used to get rid or at least plan for the highest risks. After this phase the team has a plan of what to work on and how much work is to be done. This part should take about 25% of the total time.
Construction: This step is where the team implements the functionality of the product and starts working on the development. A big part of it, is to get rid of risks which hinder the process. This is the biggest part, taking about 50% of the total time.
Transition: This final step of RUP is ued to test the product and finalize the deployment. About 20% of the time should be planned for it.

Scrum

Scrum is a framework that helps teams work together. It is most frequently used by software development teams but its principles and lessons can be applied to all kinds of teamwork. It helps teams to find solutions for complex problems with the help of planning. It consists of different roles and parts to make it work:

- Scrum Team
  - Developers
  - Product Owner
  - Scrum Master

- Scrum Events
  - The Sprint
  - Sprint Planning
  - Daily Scrum
  - Sprint Review
  - Sprint Retrospective

- Scrum Artifacts
  - Product Backlog
  - Print Backlog
  - Increment

- Scrum Values
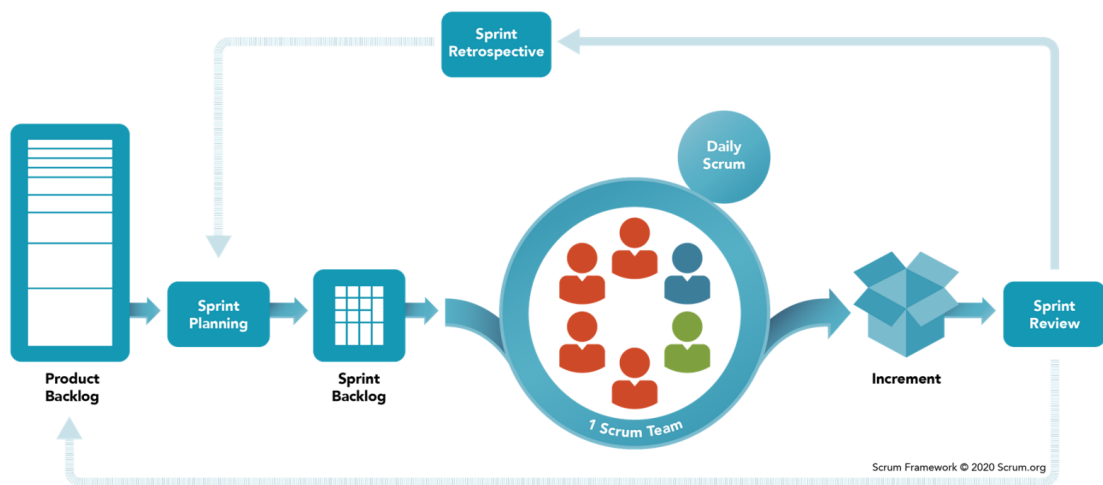 - Commitment
  - Focus
  - Openness
  - Respect
  - Courage

Figure 4.1: Scrum

4.1.3 Organisation

Participants

The "Studienarbeit"-Team consists of three students: Gianluca Nenz, Ronny Mueller and Thomas Kleb. Work on the project and documentation will be evenly distributed between these three participants. Bigger decisions are made as a team

in either the meetings with or without the advisor (the advisor will be notified on any change made).

Advisor

The teams advisor for the "Studienarbeit" is Ivan Buetler who is teaching cyber security modules at the OST.

Division of Labor

The project has multiple facets that need to be taken care of. This is why the team has decided to distributed the work load between the three. This doesn't mean that the work is done by only the chosen student but rather that he is the one responsible that it works as planned.

Gianluca Nenz
Work 1
Work 2
Work 3
Work 4

Ronny Mueller
Work 1
Work 2
Work 3
Work 4

Thomas Kleb
Work 1
Work 2
Work 3
Work 4

Table 4.2: Work Distribution per Student

4.1.4 Planning and Milestones

Phases and Iterations

The project is comprised of the four steps explained in *RUP (Rational Unified Process)*. Each of those phases has multiple iterations which create the different sprints for the project. The meetings with the advisor will be on thursdays while the team meetings will be held tuesdays. Each iteration / sprint will be of a seven day length.

We started the "Studienarbeit" before we began with the regular school. In the week before we each made research and plans about the comming project. After having a talk with the advisor it was decided to first find out the level of knowledge each student has to make it easier for the advisor to plan.

Inception			
Iteration	Start	End	Description
0	12.09.2022	18.09.2022	Collection of Ideas and planning first meeting
1	19.09.2022	25.09.2022	First meeting and handout of exercises to assess the knowledge of the students
2	26.09.2022	02.10.2022	Working on the exercises and receiving solutions for harder ones

Table 4.3: RUP: Inception Phase Planning

The elaboration phase is used to plan and assess the possible risks in this project. This consists of a documentation structure, the project plan and the risk management to make sure the construction phase has no major hickups.

Elaboration			
Iteration	Start	End	Description
3	03.10.2022	09.10.2022	First big meeting with advisor; Creating project plan and risk analysis.
4	10.10.2022	13.10.2022	Project Plan and Documentation is set; Problem Domains and Learning-concepts are defined
5	14.10.2022	25.10.2022	Lab Conepts are defined

Table 4.4: RUP: Elaboration Phase Planning

The construction phase is where the labs are primarily built.

Construction			
Iteration	Start	End	Description
6	25.10.2022	01.11.2022	—-
7	01.11.2022	08.11.2022	—-
8	08.11.2022	15.11.2022	—-
9	15.11.2022	22.11.2022	—-
10	22.11.2022	29.11.2022	—-
11	29.11.2022	06.12.2022	—-

Table 4.5: RUP: Construction Phase Planning

To make sure enough time is planned a buffer week was added to the transition phase. This phase is also mainly used to finish up the documentation and implement the different labs to Hacking Lab. The last week is used to clean up and hand in the documentation and abstract to both the OST and the advisor.

Transition			
Iteration	Start	End	Description
12	06.12.2022	13.12.2022	Buffer
13	13.12.2022	20.12.2022	—-
14	20.12.2022	23.12.2022	—-

Table 4.6: RUP: Transition Phase Planning

Milestones

To guarantee the success of the project milestones were defined with a deadline.

Milestones	Deadline	Description
M1 - Solving RE Exercises	05.10.2022	The Team solves the given exercises to find the level of RE knowledge.
M2 - Defining Problem Domains and Learnconcepts	13.10.2022	Problem Domains are defined, first Lernconcepts are planned
M3 - Lab Concepts	25.10.2022	Lab Concepts are defined to start working on the construction.
M4 - Setup Labs	06.12.2022	Labs are setup and tested.
M5 - Hand-In	23.12.2022	Document is handed in to the advisor and OST

Time Tracking

For time tracking the team has decided on using GitLabs integrated time tracking.

Issue Tracking

The issue tracking is done on GitLabs own interface to have as few difficulties as possible. To have an easier overview of the different issues the team has created tags to differentiate between the issues and their assigned student.

Meetings

The team has meetings each tuesday to elaborate problems and check up on the progress. This meetings are also used to distributed the work load and the different parts of the sprint.
On Thursdays the team meets the advisor Ivan Buetler to inform him on the progress done and the problems that came up. These meetings have different time schedules to fit everyones calender.
Each meeting will be documented and uploaded to the GIT repository. After each meeting the participants should know what to do and how to contact each other if any problems arise.

CI/CD

Testing

Projectmanagment

The whole project will use a GitLab repository. To make sure no confusion happens a multirepo principle is used where one repository is for the documentation and protocols only and other are for code, information gathered, etc. Each student works on a branch and before pushing to the main branch has another student look into the code / text written.

4.2 Risk Analysis

4.3 Project Monitoring

Overview

Milestones

Time Tracking

4.4 Personal Rapports

Gianluca Nenz

Ronny Mueller

Thomas Kleb

Chapter 5

Meetings

Nr	Phase	Date	Description
1	Elaboration	06.10.2022	Coordinate the project, documentation and ideas
2	Elaboration	13.10.2022	Present the Problemdomain with Learning Concepts and define the project plan
3	Elaboration	20.10.2022	—-
4	Construction	27.10.2022	—-
5	Construction	03.11.2022	—-
6	Construction	10.11.2022	—-

Table 5.1: Meetings held with advisor

5.1 Elaboration

Meeting 1: 06. Okt.

Deliverables: Solving the given Hacking-Lab Exercises with and without solutions to find out the level of Reverse Engineering of each student.
Discussed Topics:

- Documentation

- Goal of the Project: Hacking Lab Integration for cyber security classes.

  - Teacher needs required knowhow
  - Labs built for a student attending the third year

- Brainstorming for Problemdomains

  - Learningconcepts need to be defined for teacher and for Lab construction
  - RE Tools
  - Required Knowhow for a Reverse Engineer

- Mini Projectplan Brainstorming:

  - Analysis of Problemdomains
  - Setting up Learningconcepts
  - What is important?

- Build Lab Concepts
  - Implement with Tests

Duration: 1 hour 30 min

Meeting 2: 13. Okt.

Deliverables: Draft the Problemdomains together with Learningconcepts and the Projectplan.
Discussed Topics:

- Problemdomain Mindmap

- Learning Concepts

- What a Hacking-Lab course needs

  - Deliverables defined
  - Show which skills are trained (Mindmap / MITRA style table)
  - Exercises created as Markdown for ease of use
  - Get inspiration from already established courses

Duration: 1 hour

Meeting 3: 20. Okt.

Deliverables: Build some Lab Concepts (deliverables and overall subject), GANTT Diagramm for construction phase
Discussed Topics:
Duration:

5.2 Construction

5.3 Transition

Directory

5.6 Table Directory

5.7 Illustration Directory

Appendix

5.8 Eigenständigkeitserklärung

Eigenständigkeitserklärung

Erklärung

Wir erklären hiermit,

- dass wir die vorliegende Arbeit selbst und ohne fremde Hilfe durchgeführt haben, ausser derjenigen, welche explizit in der Aufgabenstellung erwähnt ist oder mit de Betreuer schriftlich vereinbart wurde,

- dass wir sämtliche verwendeten Quellen erwähnt und gemäss gängigen wissenschaftlichen Zitierregeln korrekt angegeben haben.

- dass wir keine duch Copyright geschützten Materialien (z.B. Bilder) in dieser Arbeit in unerlaubter Weise genutzt haben.

_____   _____
Gianluca Nenz   Date

_____   _____
Ronny Mueller   Date

_____   _____
Thomas Kleb   Date

5.9 Nutzungsrechte

5.10 Danksagung