

K.14 Fundamentals of Software Engineering

- **Course name:** Fundamentals of Software Engineering
- **Course number:** F20-FSE

K.14.1 Course Characteristics

K.14.1.1 Key concepts of the class

Scientific foundations for Software Engineering depend on the adequate use of methodologies, techniques, tools, and abstractions for developing software. This course explores fundamental techniques for producing, deploying and analyzing software systems from the Agile philosophy perspective.

K.14.1.2 What is the purpose of this course?

The main objectives of this course are the following:

- To understand agile models of interaction design, development, and project management and put those concepts in practice in real software development.
- To get acquainted with processes, technologies, and activities involved eliciting, analyzing, validating, specifying, and managing functional and non-functional requirements of software systems.
- To manage risks and decisions making based on the risk management
- To be able design software, and to write code that sticks to good practices of cohesion, coupling, modularity, reusing, and encapsulation.
- To get familiar with standard reliability techniques such as validation (testing) and verification.
- To gain practical experience with refactoring techniques to improve the quality of existing code.

K.14.1.3 Course objectives based on Bloom's taxonomy

- What should a student remember at the end of the course?

By the end of the course, the students should be able to ...

- recall Agile manifesto and principles
- recognize different Agile frameworks
- define core principles and rituals of Scrum framework
- list different types of product backlog items
- recall main concept of system testing, VCM, software architecture, technical debt

- What should a student be able to understand at the end of the course?

By the end of the course, the students should be able to ...

- discuss pros and cons of different project management tools
- describe criteria for good product backlog
- describe criteria for efficient user stories
- explain how to conduct iteration planning, review and retrospective
- discuss possible strategy of Agile implementation

- What should a student be able to apply at the end of the course?

By the end of the course, the students should be able to ...

- elicit requirements and work with requirements
- design, implement, verify and maintain efficient systems
- present work in front of stakeholders
- effectively critiquing the work of others and receiving such criticism
- work in a development team and collaborate between different teams

K.14.1.4 Course evaluation

Table K.34: Course grade breakdown

Type	Default points	Proposed points
Group work	20	60
Individual work	30	40
Test	50	10

Group work includes:

- **Sprint Delivery.** This course is organized around developing the program in four 2-weeks iterations (including final delivery). The delivery mostly focuses on the presented works that were done (User stories that were finished during the sprint).
- **Artifacts delivery.** During course, besides code delivery students will create eight artifacts (stakeholder interview report, project management tools analysis report, impact mapping, initial project backlog, a definition of done checklist, user stories acceptance criteria, risk breakdown structure, list of quality attribute scenarios) and should follow and update them.
- **Critique Report Delivery.** Students will have an opportunity to evaluate sprints and artifacts delivery of other teams and provide five critique reports that will emphasize what was done well and what needs to improve.
- **Critique Report Reflection.** Students will have an opportunity to update work that was done based on the critique reports from other team and teacher assistant comments and provide critique report reflection that explains changes.

Individual work includes:

- **Peer review report.** At the end of the course, students will sum up the contribution of the work and provide a review of the teammates. The evaluation will be based on the quality of the report (1%) creation and teammates reflection (4%).
- **Participation, presentation delivery.** Students that take participation during lecture and labs classes as well as performing during the presentation will get from 1 to 10% of the grade based on the instructor and lector's judgment.
- **Attendance.** It is mandatory to participate in the lectures, labs, and presentations (even if not presenting). Presence on all classes will give 5% to the final grade
- **Optional.** Final grades may be adjusted up to 2% taking into account factors like personal students' effort, improvements during course, and extra credit problems. This additional 2% is only based on the instructor's personal judgment, and it's not subject to further clarification.

Test.

- After the end of the first part of the course (4 weeks), there will be a test that covers all theoretical and practical material that was presented.

K.14.1.5 Grades range

Table K.35: Course grading range

Grade	Default range	Proposed range
A. Excellent	90-100	85-100
B. Good	75-89	75-84
C. Satisfactory	60-74	60-74
D. Poor	0-59	0-59

K.14.1.6 Resources and reference material

- Essential Scrum: a practical guide to the most popular agile process by Kenneth S. Rubin.
- Lean Software Development in Action by Janes, Andrea, Succi, Giancarlo
- Software Architecture in Practice, Third Edition by Rick Kazman, Paul Clements, Len Bass
- Design Patterns: Elements of Reusable Object-Oriented Software 1st Edition by Erich Gamma, Richard Helm, Ralph Johnson
- Pro Git by Ben Straub and Scott Chacon

K.14.2 Course Sections

The main sections of the course and approximate hour distribution between them is as follows:

Table K.36: Course Sections

Section	Section Title	Teaching Hours
1	Introduction	20
2	SCRUM rituals overview	20
3	Iteration 0	12
4	Iteration 1	12
5	Iteration 2	12
6	Iteration 3	12
7	Iteration 4	12
8	Conclusion	4

K.14.2.1 Section 1

Section title: Introduction

Topics covered in this section:

- Course explanation
- Transformation in IT Sphere
- Modern trends and the evolution of development processes
- Building IT teams (roles and responsibilities)
- Agile philosophy
- Projects presentation

What forms of evaluation were used to test students' performance in this section?

Form	Yes/No
Development of individual parts of software product code	0
Homework and group projects	1
Midterm evaluation	0
Testing (written or computer based)	1
Reports	1
Essays	0
Oral polls	1
Discussions	1

Typical tasks and exercises for ongoing performance evaluation within this section

Given what was discussed in class regarding how people conceptualize their experiences, design an interview script and conduct the interview with a relevant stakeholder for your class project.

1. First, discuss your class project with a stakeholder. With your target stakeholder in mind, design an interview script with closed- and open-ended questions that cover the professional background or knowledge. The questions should also explore in- depth how that person conducts their business in the problem space.
2. Second, prepare to conduct the interview. Plan for 15 minutes and record the interview using an audio recorder. Take brief notes as you go, recognizing that you have the audio recording to complement your notes.

3. Third, transcribe 1-page of information from your interview that describes the most relevant events to your project. This should include the user's goals and any explanation of how and why they pursue those goals.
4. Fourth, choose 60-second sample from audio file that cover most important information and provide (word-to-word) transcription of it.

Typical tasks and exercises for seminar classes (labs) within this section

Learn to analyze and compare different programs/tools that already on the market.

1. Create 4-pages document with comparison for tools Project management tools and mind map tools
2. Create 10 minutes presentation with tools observation.

Conduct the interview according to homework assignment

Test questions for final assessment in this section

1. What is the project management triangle?
2. What does mean challenging project?
3. What the structure of waterfall model?
4. What the difference between iterative and incremental model?
5. What the key points in scrum process model?
6. What differences between functional, cross-functional and team overlap structures?
7. What difference between semantic vs episodic memory?
8. What are the basis components of a question

K.14.2.2 Section 2

Section title: SCRUM rituals overview

Topics covered in this section:

- Agile Principles
- Introduction to Product Backlog
- Assessing the criteria's DEEP and INVEST

- Methods for Product Backlog estimation
- Definition of Done (DoD)
- Acceptance criteria (AC)

What forms of evaluation were used to test students' performance in this section?

Form	Yes/No
Development of individual parts of software product code	0
Homework and group projects	1
Midterm evaluation	1
Testing (written or computer based)	0
Reports	1
Essays	0
Oral polls	1
Discussions	1

Typical tasks and exercises for ongoing performance evaluation within this section

Product Backlog and INVEST

1. In one document: Provide an overview of your product backlog, your INVEST criteria evaluation and table with user stories that assessed with INVEST technique
2. Create slides to present your overall backlog, user stories and explain your evaluation of the user stories. Prepare for 10 minutes exhibition plus 3 minutes Q&A session.

Product Backlog estimation, Acceptance criteria, Definition of Done

1. Update your product backlog with clarification of user stories that will come to the Iteration 2.
2. Provide the estimation for your user stories in story points based on the planning poker technique. Use as the basis your team velocity for Iteration 1.
3. Apply for every user story in the product backlog the acceptance criteria in a format that was presented during the lectures.
4. Create a definition of done checklist for your project.
5. Provide brief justification of statements in your checklist.
6. Create slides to present your acceptance criteria for user stories, and your definition of the done checklist.
7. Consider to present it with your Sprint 1 deliveries. The final performance should be no more than 10 minutes plus 3 minutes Q&A session.

Typical tasks and exercises for seminar classes (labs) within this section

Product Backlog

1. Create a Product Backlog for project that was chosen for impact map creation, ask questions if have problems with creation
2. Create a table with INVEST criteria
3. Apply INVEST technique for product backlog items
4. Present the results and answer to group mate questions

Product Backlog estimation, Acceptance criteria

1. Update estimation for US and provide AC for the project that was presented last seminar classes, ask questions if have problems
2. Present the results and answer to group mate questions

Test questions for final assessment in this section

1. What means appropriate level of detail in the backlog?
2. With given user story provide its evaluation with INVEST grading criteria's.
3. Explain how Definition of Done might help to development team. What problems arises when you do not use such technique?
4. What benefits for customer of providing Acceptance Criteria for every user story? What about developers, testers?
5. Is it reasonable to provide Acceptance Criteria for epics stories?
6. When it better to use T-short size estimation, Story Points estimation and Ideal Hours (days)?

K.14.2.3 Section 3

Section title: Iteration 0

Topics covered in this section:

- System Testing
- Version Control Management
- Sprint Planning / Review /Retrospective

What forms of evaluation were used to test students' performance in this section?

Form	Yes/No
Development of individual parts of software product code	1
Homework and group projects	1
Midterm evaluation	0
Testing (written or computer based)	0
Reports	1
Essays	1
Oral polls	1
Discussions	1

Product delivery

1. Implement the tasks that were chosen in the iteration backlog from the product backlog
2. During implementation focus on the process as well and follow the scrum rituals

Critique report

1. Investigate a project report prepared by another team. Team should go over it very carefully on an individual basis, and then meet as a team to discuss conclusions.
2. The team will prepare a brief (appx. 3-5 pages) report summarizing the strengths, weaknesses and recommended actions for the team whose project report are critiquing. (Provide reasoning for artifacts help you to look at all deliveries as a single piece and refreshing the knowledge. Revise deliveries of Sprint. Conduct a code review and analyze a process of the last Sprint of the critiquing team.
3. One member of the team will deliver a 15-minute summary (presentation) of critique to the class.

Critique report reflection

1. Students receive a critique report from another team that attempts to describe the strengths and weaknesses of your artifacts. Every team should first go over their report very carefully on an individual basis and then meet as a team to discuss your conclusions.
2. Update teams project artifacts (impact map, product backlog, INVEST), project management board and source code to include any changes recommended in the critique report that your team finds acceptable.

3. Prepare a brief (appx. 2-4 pages) report summarizing the changes that were made based on the critique report, discussions in class and comments in the evaluation. If changes were not made, the explanation should be provided.

Typical tasks and exercises for seminar classes (labs) within this section

System Testing, Version Control Management

1. Examine and create unit tests, integration's tests, UI tests for your project.
2. Example of working with Git (Setup repository, connect it with students project, create master and developer brunches)

Scrum rituals discussion

1. Discuss Scrum rituals: Planning, Review, Retrospective
2. In a group create a list with positive and negative outcomes of each ritual
3. Present your list and discuss how to achieve positive outcomes and avoid negative outcomes.

Iteration delivery

1. Prepare the agenda for the meeting in advanced including questions to the stakeholder
2. Record the conversation for future analysis;
3. Analyzed Retrospective of the last Iteration, including answers at questions: What was good during the Sprint? What was bad during the Sprint? How do you decide to improve the next sprint?
4. Prepare project managing board, updated to the next Sprint;
5. Prepare project backlog and sprint backlog.

Test questions for final assessment in this section

1. Describe differences between Unit test and integration test.
2. Enumerate different types of the test.
3. Explain the purpose of different types of the test.
4. Draw the git workflow. Reason about benefits and weaknesses of presenting workflow.
5. Describe sprint rituals. Reason of necessity to follow rituals.

K.14.2.4 Section 4

Section title: Iteration 1

Topics covered in this section:

- Conceptual Data Modeling and Database Design
- Entity-Relationship (ER) Model

What forms of evaluation were used to test students' performance in this section?

Form	Yes/No
Development of individual parts of software product code	...
Homework and group projects	...
Midterm evaluation	...
Testing (written or computer based)	...
Reports	...
Essays	...
Oral polls	...
Discussions	...

Typical tasks and exercises for ongoing performance evaluation within this section

Conceptual Data Modeling and Database Design

1. Using High-Level Conceptual Data Models for Database Design
2. Entity-Relationship (ER) Model
 - (a) Entity Types, Entity Sets, Attributes, and Keys
 - (b) Relationship Types, Relationship Sets, Roles, and Structural Constraints
 - (c) Weak Entity Types
 - (d) Naming Conventions

Product delivery

1. Implement the tasks that were chosen in the iteration backlog from the product backlog
2. During implementation focus on the process as well and follow the scrum rituals

Critique report

1. Investigate a project report prepared by another team. Team should go over it very carefully on an individual basis, and then meet as a team to discuss conclusions.
2. The team will prepare a brief (appx. 3-5 pages) report summarizing the strengths, weaknesses and recommended actions for the team whose project report are critiquing. (Provide reasoning for artifacts help you to look at all deliveries as a single piece and refreshing the knowledge. Revise deliveries of Sprint. Conduct a code review and analyze a process of the last Sprint of the critiquing team.
3. One member of the team will deliver a 15-minute summary (presentation) of critique to the class.

Critique report reflection

1. Students receive a critique report from another team that attempts to describe the strengths and weaknesses of your artifacts. Every team should first go over their report very carefully on an individual basis and then meet as a team to discuss your conclusions.
2. Update teams project artifacts (impact map, product backlog, INVEST), project management board and source code to include any changes recommended in the critique report that your team finds acceptable.
3. Prepare a brief (appx. 2-4 pages) report summarizing the changes that were made based on the critique report, discussions in class and comments in the evaluation. If changes were not made, the explanation should be provided.

Typical tasks and exercises for seminar classes (labs) within this section

Conceptual Data Modeling and Database Design

1. Requirements collection and analysis using a case study or realistic scenario
2. Collect and distinguish between Functional Requirements and Data Requirements
3. Elaborate a conceptual schema using the Entity-Relationship (ER) Model

Iteration delivery

1. Prepare the agenda for the meeting in advanced including questions to the stakeholder

2. Record the conversation for future analysis;
3. Analyzed Retrospective of the last Iteration, including answers at questions:
What was good during the Sprint? What was bad during the Sprint? How do you decide to improve the next sprint?
4. Prepare project managing board, updated to the next Sprint;
5. Prepare project backlog and sprint backlog.

Test questions for final assessment in this section

1. Given a set of requirements in natural language, elaborate the respective Entity-Relationship (ER) Model and its main components
2. Enumerate the differences between Weak and normal Entities

K.14.2.5 Section 5

Section title: Iteration 2

Topics covered in this section:

- Relational Database Design by ER-to-Relational Mapping
- SQL (Basic, complex queries)

What forms of evaluation were used to test students' performance in this section?

Form	Yes/No
Development of individual parts of software product code	...
Homework and group projects	...
Midterm evaluation	...
Testing (written or computer based)	...
Reports	...
Essays	...
Oral polls	...
Discussions	...

Typical tasks and exercises for ongoing performance evaluation within this section

Relational Database Design by ER-to-Relational Mapping

1. Relational Model Concepts
2. Relational Model Constraints and Relational Database Schemes

3. Update Operations, Transactions, and Dealing with Constraint Violations

SQL (Basic, complex queries)

1. SQL Data Definition and Data Types
2. Specifying Constraints in SQL
3. Basic Retrieval Queries in SQL
4. INSERT, DELETE, and UPDATE Statements in SQL

Product delivery

1. Implement the tasks that were chosen in the iteration backlog from the product backlog
2. During implementation focus on the process as well and follow the scrum rituals

Critique report

1. Investigate a project report prepared by another team. Team should go over it very carefully on an individual basis, and then meet as a team to discuss conclusions.
2. The team will prepare a brief (appx. 3-5 pages) report summarizing the strengths, weaknesses and recommended actions for the team whose project report are critiquing. (Provide reasoning for artifacts help you to look at all deliveries as a single piece and refreshing the knowledge. Revise deliveries of Sprint. Conduct a code review and analyze a process of the last Sprint of the critiquing team.
3. One member of the team will deliver a 15-minute summary (presentation) of critique to the class.

Critique report reflection

1. Students receive a critique report from another team that attempts to describe the strengths and weaknesses of your artifacts. Every team should first go over their report very carefully on an individual basis and then meet as a team to discuss your conclusions.
2. Update teams project artifacts (impact map, product backlog, INVEST), project management board and source code to include any changes recommended in the critique report that your team finds acceptable.

3. Prepare a brief (appx. 2-4 pages) report summarizing the changes that were made based on the critique report, discussions in class and comments in the evaluation. If changes were not made, the explanation should be provided.

Typical tasks and exercises for seminar classes (labs) within this section

Relational Database Design by ER-to-Relational Mapping

1. Given the description of a business description (probably the same given for the ER diagram), identify the Domains, Attributes, Tuples, and Relations.
2. Build the relational database schema for the given business description.

SQL (Basic, complex queries)

1. Create basic database objects (tables, columns, etc)
2. Execute basic SQL queries (SELECT, INSERT, DELETE and UPDATE)

Iteration delivery

1. Prepare the agenda for the meeting in advanced including questions to the stakeholder
2. Record the conversation for future analysis;
3. Analyzed Retrospective of the last Iteration, including answers at questions: What was good during the Sprint? What was bad during the Sprint? How do you decide to improve the next sprint?
4. Prepare project managing board, updated to the next Sprint;
5. Prepare project backlog and sprint backlog.

Test questions for final assessment in this section

1. Given the description of a business description (probably the same given for the ER diagram), identify the Domains, Attributes, Tuples, and Relations.
2. Build the relational database schema for the given business description.
3. Given a relational schema, write appropriate SQL DDL statements for declaring all the database items.
4. Write SQL DML statements for basic queries and retrieve.

K.14.2.6 Section 6

Section title: Iteration 3

Topics covered in this section:

- Basics of Functional Dependencies and Normalization for Relational Databases
- Introduction to Relational Algebra

What forms of evaluation were used to test students' performance in this section?

Form	Yes/No
Development of individual parts of software product code	...
Homework and group projects	...
Midterm evaluation	...
Testing (written or computer based)	...
Reports	...
Essays	...
Oral polls	...
Discussions	...

Typical tasks and exercises for ongoing performance evaluation within this section

Basics of Functional Dependencies and Normalization for Relational Databases

1. Functional Dependencies
2. Normal Forms Based on Primary Keys
3. General Definitions of Second and Third Normal Forms
4. Boyce-Codd Normal Form
5. Multivalued Dependency and Fourth Normal Form

Introduction to Relational Algebra

1. Unary Relational Operations: SELECT and PROJECT
2. Relational Algebra Operations from Set Theory
3. Binary Relational Operations: JOIN and DIVISION

Product delivery

1. Implement the tasks that were chosen in the iteration backlog from the product backlog
2. During implementation focus on the process as well and follow the scrum rituals

Critique report

1. Investigate a project report prepared by another team. Team should go over it very carefully on an individual basis, and then meet as a team to discuss conclusions.
2. The team will prepare a brief (appx. 3-5 pages) report summarizing the strengths, weaknesses and recommended actions for the team whose project report are critiquing. (Provide reasoning for artifacts help you to look at all deliveries as a single piece and refreshing the knowledge. Revise deliveries of Sprint. Conduct a code review and analyze a process of the last Sprint of the critiquing team.
3. One member of the team will deliver a 15-minute summary (presentation) of critique to the class.

Critique report reflection

1. Students receive a critique report from another team that attempts to describe the strengths and weaknesses of your artifacts. Every team should first go over their report very carefully on an individual basis and then meet as a team to discuss your conclusions.
2. Update teams project artifacts (impact map, product backlog, INVEST), project management board and source code to include any changes recommended in the critique report that your team finds acceptable.
3. Prepare a brief (appx. 2-4 pages) report summarizing the changes that were made based on the critique report, discussions in class and comments in the evaluation. If changes were not made, the explanation should be provided.

Typical tasks and exercises for seminar classes (labs) within this section

Basics of Functional Dependencies and Normalization for Relational Databases

1. Practice on the different Normal Forms (1NF, 2NF, 3NF, BCNF and 4th NF)

Introduction to Relational Algebra

1. Practice on the different operators of Relational Algebra

Iteration delivery

1. Prepare the agenda for the meeting in advanced including questions to the stakeholder
2. Record the conversation for future analysis;
3. Analyzed Retrospective of the last Iteration, including answers at questions: What was good during the Sprint? What was bad during the Sprint? How do you decide to improve the next sprint?
4. Prepare project managing board, updated to the next Sprint;
5. Prepare project backlog and sprint backlog.

Test questions for final assessment in this section

1. Given business requirements in natural language, present the different Normal Forms (1NF, 2NF, 3NF, BCNF and 4th NF)
2. Given ad-hoc business queries, describe them in terms of Relational Algebra
3. Relate Relational Algebra queries and SQL

K.14.2.7 Section 7

Section title: Iteration 4

Topics covered in this section:

- Technical Debt
- SOLID technique

What forms of evaluation were used to test students' performance in this section?

Form	Yes/No
Development of individual parts of software product code	1
Homework and group projects	1
Midterm evaluation	0
Testing (written or computer based)	0
Reports	1
Essays	1
Oral polls	1
Discussions	1

Typical tasks and exercises for ongoing performance evaluation within this section

Product delivery

1. Implement the tasks that were chosen in the iteration backlog from the product backlog
2. During implementation focus on the process as well and follow the scrum rituals

Final product delivery

1. Implement the tasks that were chosen in the iteration backlog from the product backlog be sure that you have MVP of the product
2. During implementation focus on the process as well and follow the scrum rituals
3. Provide peer review based on the survey

Typical tasks and exercises for seminar classes (labs) within this section

Reflection

1. Provide the feedback based on the structure of the course, artifacts, deliveries
2. Reason what material you think you may use in the future
3. What problems you may address with new knowledge?

Typical tasks and exercises for seminar classes (labs) within this section

Technical Debt, SOLID technique

1. Install Sonar Cube and connect it with your project code.
2. Examine the Sonar Cube reports.
3. Reason about your Technical Debt. Was it done intentionally or unintentionally.
4. Discuss your previous experience and/or situation that might happen. Think when the usage of Technical Debt is justified and when it is not.

Test questions for final assessment in this section

1. Enumerate type of Technical Debt
2. Do we always have to pay technical debt?
3. How fast should we pay technical debt?
4. Reason about the situation when software is rarely modified, should we pay technical debt?
5. What is the relationship between product support and technical debt?

6. Explain what acronym SOLID means. Describe two out of for best SOLID practices.

Teammates evaluation

1. Asses the contribution to create/improve artifacts
2. Asses the contribution to prepare presentations
3. Asses the contribution in Sprints (developing software)
4. Asses the contribution to create critique reports
5. Asses the contribution to create critique reports reflections
6. Asses the level of conflicts
7. Asses the difficulties of managing person
8. Highlight the areas of improvement (in general)