Module Description, available in: EN

Software Engineering and Architectures

General Information

Number of ECTS Credits

3

Module code

TSM_SoftwEng

Valid for academic year

2025-26

Last modification

2023-11-13

Coordinator of the module

Sandy Ingram (HES-SO, sandy.ingram@hefr.ch)

Explanations regarding the language definitions for each location:

- Instruction is given in the language defined below for each location/each time the module is held.
- Documentation is available in the languages defined below. Where documents are in several languages, the percentage distribution is shown (100% = all the documentation).
- The examination is available 100% in the languages shown for each location/each time it is held.

	Lausanne			Lugano	Zurich		
Instruction	**X** E 100%				**X** E 100%		
Documentation	**X** E 100%				**X** E 100%		
Examination	**X** E 100%				**X** E 100%		

Module Category

TSM Technical scientific module

Profile Recommendations

This module is recommended for the following profiles:

	Lausanne	Zurich
Profile	CS	CS

Lessons

2 lecture periods and 1 tutorial period per week

Entry level competences

Prerequisites, previous knowledge

- Object-oriented programming and design in more than one programming language
- Unified Modeling Language (UML)
- Design Patterns: Elements of Reusable Object-Oriented Software (Gamma, Helm, Johnson, Vlissides; ISBN 0-201-63361-2)
- Version and configuration management concepts
- Unit testing concepts and practice

Brief course description of module objectives and content

The module provides an in-depth view of selected topics of modern software engineering. These stem from the fields: modern software development processes, software architecture, and the principles of evolution of software systems.

Aims, content, methods

Learning objectives and competencies to be acquired

- The student applies and understands benefits and liabilities of agile and lean development
- The student knows about advanced architectural and design patterns and uses them to drive and reflect on design decisions
- The student has the awareness of software as a continuously evolving and complex system
- The student knows and can select maintenance and evolution techniques for continuous development of evolving and extension of legacy software systems while maintaining its quality

Module content with weighting of different components

Software Architectures (20%)

- Role of Software Architecture and Software Architects
  - reference models, reference architectures
  - architectural structures and views
  - software architecture documentation
- Advanced Design Concepts
  - Design by contract
  - The SOLID principles
- Architecture Styles
- Architecture Patterns (vs design patterns)
- Research in Software Architecture

Modern Software Engineering (50%)

- Agile Principles and Practices
  - effective communication among stakeholders
  - project retrospectives, feedback techniques
  - ongoing requirements, solicitation and management
- Agile Methodologies
- Automation in Agile Software Release Management
  - Testing approaches and types
  - Agile testing quadrants
  - CI/CD Continuous Integration and Continuous Deployment (with Docker)
  - Automated testing and CI/CD Practices for Embedded Systems and Web Applications

Software Evolution (30%)

- Principles of Software Evolution
  - development, maintenance, evolution
  - software aging
  - Program comprehension
- Software Quality & Analysis
  - software quality metrics
  - software visualization
  - continuous quality control
- Evolution of Legacy Code
  - “Re”-Techniques: Reverse Engineering, Re-Engineering, Re-Factoring
  - object-oriented re-engineering
  - Testing legacy systems
- Research in Software Evolution

Teaching and learning methods

Self-study, homework assignments.

Part of the module provides pre-recorded inputs to enable asynchronous learning combined with in-class labs and exercises. Part of the labs and assignments use a provided kit based on Raspberry Pi Pico-W (RP2040) with VS Code and provided software (C).

Literature

1. Mary & Tom Poppendiek: Lean Software Development
2. Kent Beck: eXtreme Programming Explained 2nd Ed.
3. Ken Schwaber et al, Agile Software Development with Scrum, Prentice Hall, 2002
4. Alistair Cockburn: Agile Software Development
5. Robert C. Martin: Agile Software Development
6. Tom Mens: Software Evolution
7. Doug Schmidt et.al.: Pattern-oriented Software Architecture, Vol. 2
   Frank Buschmann et al: Pattern-oriented Software Architecture, Vol. 4
8. Len Bass, Paul Clements, Rick Kazman: Software Architecture in Practice 2nd Ed.
9. Gernot Starke: Effektive Software Architekturen 2. Auflage

10. Lehmann "Laws of Software Evolution Revisited"
11. Martin Fowler et al, Refactoring
    Joshua Kerievsky, Refactoring to Patterns
12. Michael Feathers, Working Effectively with Legacy Code
13. Andreas Zeller: Why Programs Fail ISBN 1558608664

Assessment

Additional performance assessment during the semester

The module contains additional performance assessment(s) during the semester. The achieved mark of the additional performance assessment(s) applies to both the regular and the resit exam.

Description of additional performance assessment during the semester

- Corrected individual homework assignments account for 6% of the final grade.

-A graded group project accounts for 24% of the final grade and requires applying various concepts covered in class to the project planning, design, implementation, and testing phases. The group project involves both software components and embedded systems.

Basic principle for exams

As a rule, all standard final exams are conducted in written form. For resit exams, lecturers will communicate the exam format (written/oral) together with the exam schedule.

Standard final exam for a module and written resit exam

Kind of exam

Written exam

Duration of exam

120 minutes

Permissible aids

Aids permitted as specified below:

Permissible electronic aids

No electronic aids permitted

Other permissible aids

A personal summary of max. 20 pages A4 (or 10 sheets) on paper, with the student name on it. It does not have to be handwritten, but the summary content must be personal, and not a copy of someone else. The summary will be collected with the exam and not returned.

Exception: In case of an electronic Moodle exam, adjustments to the permissible aids may occur. Lecturers will announce the final permissible aids prior to the exam session.

Special case: Resit exam as oral exam

Kind of exam

Oral exam

Duration of exam

30 minutes

Permissible aids

No aids permitted