



ID2207 Moderna metoder inom programvaruutveckling (software engineering) 7,5 hp

Modern Methods in Software Engineering

Fastställande

Kursplan för ID2207 gäller från och med VT19

Betygsskala

A, B, C, D, E, FX, F

Utbildningsnivå

Avancerad nivå

Huvudområden

Datalogi och datateknik

Särskild behörighet

- Computer Science courses 30 hp
- Operating Systems courses 7,5 hp
- Computer Programming courses 7,5 hp
- English "level B" (Swedish Gymnasium)

Undervisningsspråk

Undervisningsspråk anges i kurstillfällesinformationen i kurs- och programkatalogen.

Lärandemål

The course aims both in giving students knowledge about modern software development methods and developing skills in usage the methods.

Our goal is to present a variety of approaches to software development and discuss their applicability boundaries, benefits, restriction and complementariness.

During the course students should learn about Software Engineering methods. In particular, they:

1. Learn methods for dealing with complexity and changes in software construction. This means that students should get understanding of main approaches to abstraction, models, decomposition and software life-cycle.
2. Understand basic components of software development process. This means that students should learn main methods and approaches to requirement elicitation and analysis, system and object design.
3. Learn some modern approaches to software development. This means that students should learn about agile methods of software development and have experience in applying them to desing a software system.
4. To get experience in evaluating different methods for producing of a high quality software system within time. This means that students should get practice in applying and comparison of different approaches to software development.
5. To get experience in reporting and discussing results of the course homework and project both in oral and written forms.
6. Understand ethical aspects and importance of sustainability in software development.

The course also includes a seminar as a part of the Software Engineering of Distributed Systems master program. The intention of the seminar is to put the course into the context of the software engineering research in general and into the context of the master program in particular.

Kursinnehåll

Introduction and basic concepts of Software Engineering (SE). Abstraction/Models and Decomposition. Software Life-Cycle. Unified process. Software Modeling language. Unified Modeling Language (UML). Requirements elicitation and analysis. System design. Object design. Applying patterns. Refactoring. Mapping models to code. Testing. Agile software development and agile modeling. Basics of Extreme Programming. Software project management.

Practical part of the course includes exercises and a small software development project applying SE methods.

Kurslitteratur

Textbook for the course:

Object-Oriented Software Engineering: Using UML, Patterns and Java: International Edition, 3/E, Bernd Bruegge, Allen H. Dutoit, ISBN: 0136061257, Publisher: Prentice Hall, Copyright: 2010, Format: Paper; 800 pp Published: 29 July 2009 (available in the Kista Electrum book store)

Lecture notes

Recommended Reading:

The following sources are recommended to obtain a deeper understanding of the subject.

- E. Gamma et al. Elements of Reusable Object-Oriented Software. Addison-Wesley, 1995
- M. Shaw, D. Garlan. Software Architecture. Perspectives on an Emerging Discipline. Prentice-Hall, 1996
- <http://www.extremeprogramming.org/start.html>
- Kent Beck. Extreme Programming Explained: Embrace Change, Publisher: Addison-Wesley Professional; 1st edition (October 5, 1999)
- Martin Fowler, Kent Beck, John Brant, William Opdyke, Don Roberts. Refactoring: Improving the Design of Existing Code, Addison-Wesley Professional; 1st edition (June 28, 1999)
- M. Matskin and E. Tyugu. Structural Synthesis of Programs and Its Extensions. Computer and Informatics Journal, v. 20, 2001, pp. 1-25

Additional articles in the curriculum may be added during the course.

Examination

- ANN1 - Hemprojekt, 3,0 hp, betygsskala: P, F
- TEN1 - Tentamen, 4,5 hp, betygsskala: A, B, C, D, E, FX, F

Examinator beslutar, baserat på rekommendation från KTH:s handläggare av stöd till studenter med funktionsnedsättning, om eventuell anpassad examination för studenter med dokumenterad, varaktig funktionsnedsättning.

Examinator får medge annan examinationsform vid omexamination av enstaka studenter.

När kurs inte längre ges har student möjlighet att examineras under ytterligare två läsår.

Written examination (TEN1 4,5 hp)

Etiskt förhållningssätt

- Vid grupparbete har alla i gruppen ansvar för gruppens arbete.

- Vid examination ska varje student ärligt redovisa hjälp som erhållits och källor som använts.
- Vid muntlig examination ska varje student kunna redogöra för hela uppgiften och hela lösningen.