KTH Microelectronics
and Information Technology

Exam in ID2207 Modern Methods in Software Engineering, 2009-10-22, 14:00-18:00

Rules

This exam is “closed book” and you are not allowed to bring any material or equipment (such as laptops, PDAs, or mobile phones) with you. The only exceptions are English to “your favorite language” dictionary and pencils.

Instructions

- Please read the entire exam first!
- Write clearly
- Each sheet of paper must contain your name, ”personnummer”, Problem number and a unique sheet number
- Write only on one page of a sheet. Do not use the back side
- Sort your sheets according to the problem’s numbering
- Only one Problem must be reported on each sheet
- If more than one sheet is needed the continuation should be clearly noted on the beginning of each sheet and the sheet numbers used should be consecutive
- Always motivate your answers. Lack of clearly stated motivation can lead to a reduction in the number of points given
- The tasks are not necessarily sorted in order of difficulty. If you get stuck it might be a good idea to go on to the next task.

Grading

The grades depend on the sum of exam and bonus points n:

$n < 50$ fail (F)
$50 \le n < 60$ grade E
$60 \le n < 70$ grade D
$70 \le n < 80$ grade C
$80 \le n < 90$ grade B
$90 \le n$ grade A

GOOD LUCK!

Mihhail Matskin, mobile 0704614269

Problem I. General questions

a) Briefly explain difference between functional and object oriented decomposition. What are pros and cons of each of them?

(5p)

b) Why do we apply principle of Falsification in software engineering?

(4p)

Problem II. Software Life Cycle and UML

a) Draw a UML class diagram representing Unified Software Development Process (UP)?

(6p)

b) We considered 3 formal software development models: inductive, deductive and transformational. Briefly (without technical details) explain main features of each of these models.

(6p)

c) What is Model Driven Architecture?

(5p)

Problem III. UML and OOP

a) Explain difference between Concept and Phenomena. Give an example of Concept and Phenomena in software.

(4p)

b) Using only one class draw a class diagram which states that a course has successor(s) and prerequisite(s) and if two courses cover nearly the same material, taking one of them may prevent a student from taking the other and vice versa.

(6p)

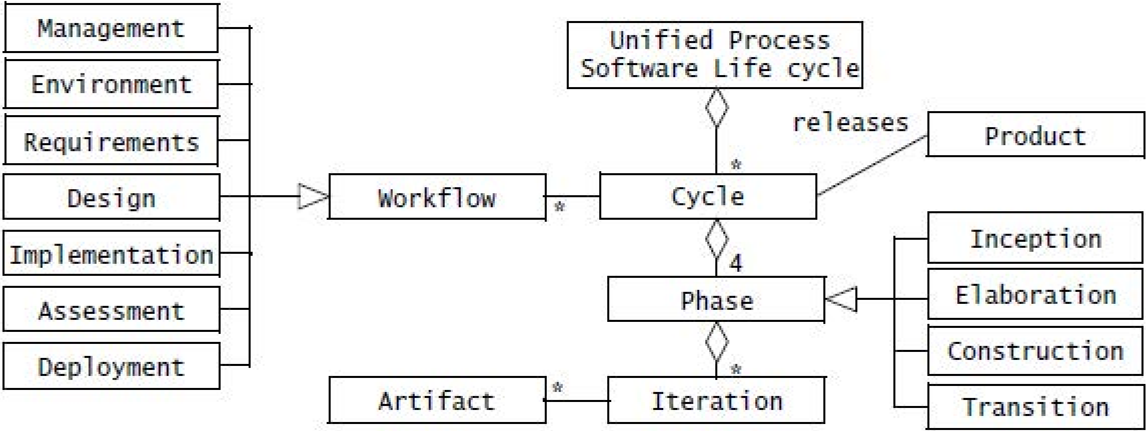
Problem IV. Requirements Elicitation

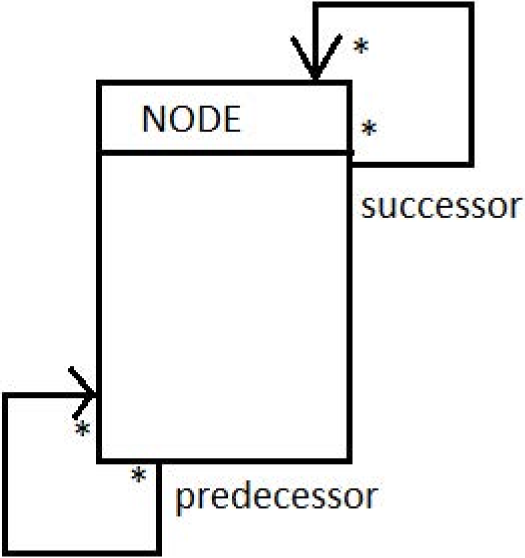
a) What is difference between traditional and modern methods for requirements elicitation? Where each of these methods is more appropriate?

(5p)

b) We identified the following types of scenario: As-Is, Visionary, Evaluation and Training scenario. Briefly explain each of them.

(4p)


Management
Environment
Requirements
Design
Implementation
Assessment
Deployment
Workflow
*
Unified Process
Software Life cycle
*
releases
Product
Cycle
4
Phase
Inception
Elaboration
Construction
Transition
*
Artifact
*
Iteration


NODE
*
*
successor
*
*
predecessor

Problem V. Requirements Analysis

a) What are at least three differences between the analysis model and the design model of a software product?

(4p)

b) Which restrictions of Object-oriented programming can be resolved in Aspect oriented programming?

(5p)

Problem VI. System Design

a) Explain the Pipe and Filter architectural style. For which systems it is suitable and for which systems it is not suitable?

(5p)

b) Explain the following terms: coupling, cohesion, partitioning and layering.

(4p)

Problem VII. Object Design - Reuse

a) Briefly explain in which cases it is reasonable to use each of the following design patterns: Adapter, Bridge, Proxy and Strategy.

(6p)

b) If you are given a choice of using implementation inheritance, delegation or specification inheritance for reuse how do you put priorities among them for usage? Briefly explain.

(5p)

Problem VIII. Object Design – Interface design

a) Assume that you have the following 3 classes: parallelogram, rhombus and square. Assume that rhombus inherits from parallelogram and square inherits from rhombus. Write invariants in OCL for each of these classes taking into account the inheritance of contracts.

(5p)

Problem IX. Moving to Code

a) There are 2 different ways of mapping inheritance into the relational database tables: with a separate table and by duplicating columns. Explain these ways with examples and compare them in terms of modifiability and response time.

(6p)

b) If we consider forward engineering then what is a transformation that we can apply to map class diagram into code?

(5p)

Problem X. Testing.

a) What is a goal of partitioning into equivalence classes in black-box testing? What are guidelines for selection of equivalence classes?

(5p)

Problem XI. Agile Software Development.

a) What is metaphor in Extreme programming? Give an example.

(5p)

-----------------------------End of Exam---