T–76.3601, Introduction to Software Engineering
Mid-term Exam 1, Take 1, 3.3.2006

Instructions:
- Write your name, student number, degree program and signature in the reserved space below
- Write your name and student number at the bottom of *each sheet*
- Answer the questions in the spaces provided on the question sheets. If you run out of room for an answer, continue on the back of the page.
- *You can answer in English, Finnish or Swedish.*

Ohjeita:
- Kirjoita nimesi, opiskelijanumerosi, koulutusohjelmasi sekä allekirjoituksesi alla olevaan tilaan
- Kirjoita nimesi ja opiskelijanumerosi jokaisen paperin alareunaan
- Vastaa kysymyksiin koepaperissa varattuun tilaan. Jos tarvitset lisätilaa, kirjoita paperin takapuolelle.
- Voit vastata englanniksi, suomeksi tai ruotsiksi.

Instruktioner:
- Skriv ditt namn, studentnummer, utbildningsprogram samt underskrift i det reserverade utrymmet nedan
- Skriv ditt namn och studentnummer nere på varje ark
- Besvara frågorna i utrymmet på provpappren. Om du behöver mera utrymme kan du skriva på arkets baksida.
- Du kan svara på engelska, svenska eller finska.

1. Define the following terms. *Määrittele seuraavat termit. Definiera följande termer.*

   (a) Software Engineering. *Ohjelmistotuotanto. Programvaruproduktion.* (1p)

   Solution: One of the following:

   - The establishment and use of sound engineering principles in order to obtain economically software that is reliable and works efficiently on real machines.
   - IEEE:
     1. The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software, that is, the application of engineering to software
     2. The study of approaches in (1)

   (b) Agile Software Development. *Ketterä ohjelmistokehitys. Vig (agil) programvaruutveckling.* (1p)

   Solution: An approach to software development that combines a philosophy and a set of development guidelines. The philosophy encourages customer satisfaction and early incremental delivery of software; small, highly motivated project teams; informal methods; minimal software engineering work products; and overall development simplicity. The development guidelines stress delivery over analysis and design, and active and continuous communication between developers and customers.

(c) Use-case. *Käyttötapaus. Use-case.* (1p)

Solution: A sequence of actions that provide value to actors. UML defined use-case diagrams, in which use-cases are drawn as ellipses, and actors as stick figures. The relationships between actors and use-cases are drawn as lines.

(d) Regression testing. *Regressiotestaus. Regressionstestning.* (1p)

Solution: The testing of a piece of software after changes have been made to it to ensure that the changes did not result in unexpected problems. Can be done manually or by re-executing (a subset) of test cases. Not the same as *retesting*, which test that a particular problem that has been fixed has disappeared. Use case-diagrams often depict the system boundary using box(es).

(e) CMMI. (1p)

Solution: A process meta-model that is predicated on a set of system and software engineering capabilities that should be present as organizations reach different levels of process maturity and capability. Contains both a stegd and a continuous model. Used for process assessment and improvement.

(f) Smoke testing. *Savutestaus. Röktestning.* (1p)

Solution: An integration testing approach that is based upon a set of (typically automated) tests that are run for every new build of the software (typically daily).

Name/Nimi/Namn:____________________________

Student number/Opiskelijanumero/Studentnummer:____________________

2. Are the following statements true or false? Mark a **T** for true statements, and an **F** for false ones. If you don't know, leave the row empty. You will get +1 p for a correct answer, 0p for an empty answer, and -1 p for an incorrect answer. The maximum score for this question is 10, and the minimum 0, i.e., you will not get a negative score even if you have more incorrect than correct answers.

   *Ovatko seuraavat väittämät oikeita vai vääriä? Kirjoita **O** oikean väittämän ja **V** väärän väittämän jälkeen. Jos et tiedä, jätä vastaamatta. Oikeasta vastauksesta saat +1 p, tyhjästä 0p ja väärästä -1 p. Tehtävän maksimipistemäärä on 10 ja minimi 0, eli et saa negatiivista pistemäärää vaikka sinulla olisi enemmän vääriä kuin oikeita vastauksia.*

   *Är följande påståenden sanna eller falska? Skriv **S** efter de som är sanna och **F** efter de som är falska. Om du inte vet, lämna tomt. Du får +1 p för korrekt svar, -1 p för fel svar, och 0p för tom lucka. Maximipoängen för denna uppgift är 10, och minimi 0. Du kan alltså inte få negativa poäng fastän du skulle ha flera inkorrekta än korrekta svar.*

   (a) A central theme in the waterfall model is risk management.
   Vesiputousmallin keskeinen teema on riskien hallinta.
   Riskhantering är ett centralt tema i vattenfallsmodellen.

   (a) ______

   (1p)

   Solution: False

   (b) Software configuration management is solely concerned with the management of different versions of software development artifacts.
   Ohjelmistokonfiguraationhallinta keskittyy pelkästään eri ohjelmistoartefaktien versioiden hallintaan.
   Programvarukonfigurationskontroll innebär endast hantering av olika versioner av programvaruartifakter.

   (b) ______

   (1p)

   Solution: False

   (c) The term software as used in the field of software engineering refers only to the machine readable code produced by, e.g., a compiler.
   Termi ohjelmisto ohjelmistotuotannossa viittaa ainoastaan esim. kääntäjän tuottamaan koneluettavaan koodiin.
   Termen software inom programvaruproduktion refererar endast till den maskinläsbara koden som produceras t.ex. av en kompilator.

   (c) ______

   (1p)

   Solution: False

   (d) A good modularization has high cohesion and low coupling.
   Hyvässä moduulijaossa on korkea koheesio ja matala modulien välisten kytkösten taso.
   En god modularisering har hög kohesion och låg koppling.

   (d) ______

   (1p)

   Solution: True

Name/Nimi/Namn: ______________________________

Student number/Opiskelijanumero/Studentnummer: ______________________________

(e) Software project planning should be done only at the outset of the project, not later.
Ohjelmistoprojektin suunnittelu on aktiviteetti, joka suoritetaan ainoastaan projektin alussa.
Projektplanering bör göras endast i början av projektet.

(e) ______

(1p)

Solution: False

(f) Using a waterfall life-cycle model makes it easy to react to requirement changes late in the project.
Vesiputousmallin käytöllä voidaan helposti reagoida vaatimusmuutoksiin, jotka tulevat projektin myöhäisessä vaiheessa.
Genom att använda vattenfallsmodellen kan man göra det lätt att reagera på kravförändringar som kommer in i ett sent skede av projektet.

(f) ______

(1p)

Solution: False

(g) Alpha testing involves a large number of end-users.
Ohjelmiston alfatestaukseen osallistuu suuri määrä loppukäyttäjiä.
Alfatestning involverar ett stort antal slutanvändare.

(g) ______

(1p)

Solution: False

(h) PSP is a process improvement methodology targeting individual software engineers.
PSP on yksittäisiin ohjelmistoinsinööreihin keskittyvä prosessinparannusmenetelmä.
PSP är en processförbättringsmetod som koncentrerar sig på den enskilda programingenjören.

(h) ______

(1p)

Solution: True

(i) eXtreme Programming (XP) is an agile software development methodology especially suitable for well-defined and large projects.
eXtreme Programming (XP) on ketterä ohjelmistokehitysmenetelmä, joka soveltuu erityisesti hyvin määriteltyjen, isojen ohjelmistoprojektien käyttöön.
eXtreme Programming (XP) är en vig (agile) programvaruutvecklingsmetod som speciellt väl passar för väldefinierade och stora utvecklingsprojekt.

(i) ______

(1p)

Solution: False

Name/Nimi/Namn: ______________________________

Student number/Opiskelijanumero/Studentnummer: ______________________

(j) Function points is a function-oriented metric which is independent of the programming language being used.
Toimintopiste on ohjelmiston toimintoihin liittyvä mittari joka on riippumaton käytössä olevasta ohjelmointikielestä.
Funktionspoäng (function points) är mätare av ett programs funktionalitet som är oberoende av använt programmeringsspråk.

(j) ______

(1p)

Solution: True

Name/Nimi/Namn:______________________________

Student number/Opiskelijanumero/Studentnummer:______________________________

3. Why is it hard to achieve perfect quality in software engineering? Discuss various approaches to improving software quality.

   Miksi on vaikeaa saavuttaa täydellistä laatua ohjelmistotuotannossa? Kuvaile eri lähestymistapoja ohjelmistojen laadun parantamiseksi.

   Varför är det svårt att uppnå perfekt kvalitet i programvaruutveckling? Beskriv olika tillvägagångssätt för att förbättra programkvalitet. (7p)

Solution:

- Reasons (3p for well motivated answer, including two or three of the following). Other well motivated answers accepted as well.
  - Defining quality: Quality is multi-faceted, and stakeholders have different views of what perfect"quality is.
  - Uniqueness: software development is development, not manufacturing — each project is a new "experiment"
  - Software development is relient on humans — and humans tend to make mistakes
  - Software is inherently complex, subject to changes, must conform to others and invisible (Up to 1 point for all Brooks' arguments)
  - Method scalability. Software development methods that work for small projects/product does not necessarily scale up well.
  - Time and resource constraints
  - Difficulties in determining the requirements, requirements changes
- Approaches (1p each, max 4). The maximum for describing process improvement practices only is two points. Other well motivated answers accepted as well.
  - Select a suitable process model for the project
  - Improve the overall software process, e.g. using the CMM(I) or SPICE frameworks
  - Design and implementation heuristics — e.g. use modularization, information hiding, etc.
  - Testing
  - Inspections and reviews
  - Use of design methodologies, like the UML
  - Systematic and planned reuse
  - Good management
  - Active communication with customer (customer involvement)
  - Use of formal methods (when feasible)

Name/Nimi/Namn:______________________________

Student number/Opiskelijanumero/Studentnummer:______________________________

4. What are the basic tasks of requirements engineering? Explain each briefly.
Mitkä ovat vaatimusten hallinnan perustoiminnot? Kuvaile jokaista lyhyesti.
Vilka är de grundläggande funktionerna i kravhantering? Beskriv varje funktion kort. (7p)

Solution: The basic tasks of requirements engineering are: ($\frac{1}{2}$p each for correct name, $\frac{1}{2}$p each for correct explanation)

- **Inception**. Software engineers ask a set of context-free like:
  - Who is behind the request for this work?
  - Who will use the solution?
  - What are the economic benefits of a successful solution?
  - Is there another source for the solution that you need?

  The purpose is to identify all stakeholders as well as to establish preliminary feasibiliy.
- **Elicitation**. In the requirements elicitation stage, requirements are elicited from the identified stakeholders; determining what the objectives are, what is to be accomplished, how the system fits into the needs of the business and into the day-to-day work.
- **Elaboration**. The information obtained in the previous phases is expanded and refined. Focus is on developing a refined techincal model of software functions, features and constraints.
- **Negotiation**. The reconciling of requirements and resource conflicts.
- **Specification**. A description of the functionality and performance (functional and non-functional requirements) of the system to be built, as well as of constraints is produced.
- **Validation**. The requirements specification is examined to ensure that the requirements are unambiguously, consistently, complete and correctly. In practice, this is extremely hard.
- **Management**. The activity of dealing with requirements changes as the project unfolds. In many ways similar (and linked to) SCM activities.

Name/Nimi/Namn:

Student number/Opiskelijanumero/Studentnummer: