



**T-76.3601, Introduction to Software Engineering
Mid-term Exam 1, Take 2, 8.3.2008**

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 kääntöpuolelle.
- 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.

Question:	1	2	3	4	Total
Points:	6	8	8	8	30
Score:					

Name/Nimi/Namn: _____

Student number/Opiskelijanumero/Studentnummer: _____

Degree program/Koulutusohjelma/Utbildningsprogram: _____

Signature/Allekirjoitus/Underskrift: _____

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) Requirements engineering. *Vaatimusten hallinta. Kravhantering.*

(1p)

Solution: The process/workflow in software engineering dealing with requirements, i.e. what software to build.

Wikipedia:

In systems engineering and software engineering, requirements analysis encompasses those tasks that go into determining the needs or conditions to meet for a new or altered product, taking account of the possibly conflicting requirements of the various stakeholders, such as beneficiaries or users. Systematic requirements analysis is also known as requirements engineering[1]. It is sometimes referred to loosely by names such as requirements gathering, requirements capture, or requirements specification. The term requirements analysis can also be applied specifically to the analysis proper (as opposed to elicitation or documentation of the requirements, for instance).

(c) Beta testing. *Beetatestaus. Beta testning.*

(1p)

Solution: A late stage of software testing, in which it is shipped to larger numbers of end-users for testing. The purpose is typically to test the software in diverse environments and usage situations.

(d) Waterfall model. *Vesiputousmalli. Vattenfallsmodell.*

(1p)

Solution: A software life-cycle model consisting of a set of sequential steps in which software is developed. Typical phases include: requirements, specification, design, implementation, testing, and delivery

(e) Scrum.

(1p)

Solution: An agile development approach.

(f) Analysis model. *Analyysimalli. Analysmodell.*

(1p)

Solution: A high-level model of the software system that acts as a bridge between requirements and the detailed design model. Typically contains scenario-based, flow-oriented, class-based, and behavioral elements.

Name/Nimi/Namn: _____

Student number /Opiskelijanumero /Studentnummer: _____

2. Are the following statements true or false? Mark an X in the box marked **True** for true statements, and an X in the box marked **False** 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 8, 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? Rastita ruutu **True** oikean väittämän ja ruutu **False** väärän väittämän jälkeen. Jos et tiedä, jätä vastaamatta. Oikeasta vastauksesta saat +1 p, tyhjästä 0p ja väärästä -1 p. Tehtävän maksimipistemäärä on 8 ja minimi 0, eli et saa negatiivista pistemäärää vaikka sinulla olisi enemmän vääriä kuin oikeita vastauksia.*

*Är följande påståenden sanna eller falska? Skriv ett X i rutan **True** efter de som är sanna och ett X i rutan **False** 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. Maximipöängen för denna uppgift är 8, och minimi 0. Du kan alltså inte få negativa poäng fastän du skulle ha flera inkorrekta än korrekta svar.*

- (a) Agile software development models are suitable to small projects in which it is difficult to fix the requirements at the start of the project.

Ketterät ohjelmistokehitysmallit soveltuvat pieniin projekteihin, joiden vaatimuksia on vaikeaa lyödä lukkoon projektin alussa.

Viga (agile) programvaruutvecklingsmodeller passar för små projekt i vilka det är svårt att slå fast kraven i början av projektet.

True False

(1p)

Solution: True

- (b) In practice, testing cannot show the absence of bugs, only their presence.

Testaamalla voidaan käytännössä ainoastaan todeta virheiden (bugien) olemassaolo, ei niiden puuttumista. I praktiken kan man genom testning endast påvisa existensen av fel (buggar), inte avsaknaden av dem.

True False

(1p)

Solution: True

- (c) Requirements need to be elicited from all relevant stakeholder groups, not only from end-users.

Vaatimuksia pitää kerätä kaikilta olennaisilta sidosryhmiltä, ei ainoastaan loppukäyttäjiltä.

Krav bör insamlas av alla viktiga intressenter, inte endast av slutanvändarna.

True False

(1p)

Solution: True

- (d) Software project effort estimation becomes increasingly easier as the project progresses.

Ohjelmistoprojektin työmäärän arviointi muuttuu yhä helpommaksi projektin edetessä.

Estimering av arbetsmängden i ett programvaruprojekt blir lättare ju längre projektet framskrider.

True False

(1p)

Name/Nimi/Namn: _____

Student number /Opiskelijanumero /Studentnummer: _____

Solution: True

(e) Revisions and variants are different types of versions.

Revisiot ja variantit ovat versioiden eri tyyppiä.

Revisioner och varianter är olika typers versioner.

True

False

(1p)

Solution: True

Name/Nimi/Namn: _____

Student number/Opiskelijanumero/Studentnummer: _____

- (f) The term software as used in the field of software engineering refers not only to the machine readable code produced by, e.g., a compiler, but also to related documentation.

Ohjelmistotuotannossa termi ohjelmisto ei viittaa ainoastaan esim. kääntäjän tuottamaan koneluettavaan koodiin, vaan myös esimerkiksi siihen liittyvään dokumentaatioon.

Termen software inom programvaruproduktion refererar inte endast till den maskinläsbara koden som produceras t.ex. av en kompilator, utan också t.ex. till därtill hörande dokumentation.

True False

(1p)

Solution: True

- (g) Alpha testing involves a small number of select end-users.

Ohjelmiston alfatestaukseen osallistuu pieni määrä valittuja loppukäyttäjiä.

Alfatestning involverar ett litet antal utvalda slutanvändare.

True False

(1p)

Solution: True

- (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.

True False

(1p)

Solution: True

Name/Nimi/Namn: _____

Student number /Opiskelijanumero /Studentnummer: _____

3. Agile software development.

*Ketterä ohjelmistokehitys.**Vig (agil) programvaruutveckling.*

(8p)

Solution: Scoring:

- Agiligy: effective response to change 1p
- Agile manifesto, $\frac{1}{2}$ p for each correct point, total 2p
- Example models, e.g. Scrum and XP 2p
- Benefits: $\frac{1}{2}$ p each, max. 1p
- Drawbacks: $\frac{1}{2}$ p each, max 1p
- Applicability: small, non-critical projects using experienced developers 1p

Name/Nimi/Namn: _____

Student number /Opiskelijanumero /Studentnummer: _____

4. Software project effort estimation: problems and methods

*Ohjelmistoprojektin työmäärän arviointi: ongelmat ja menetelmät**Estimering av arbetsmängd i programvaruprojekt: problem och metoder*

(8p)

Solution:

- Problems: 1p each (or other reasonable problem), max 4p
 - Lack of adequate information upon which to base the estimates
 - Lack of good historical information
 - Sloppily done estimates
 - Bad attitudes towards estimation (not followed, respected or trusted)
 - Estimates are not updated as more information becomes available
- Methods: 1p each, max 4p
 - Algorithmic models, such as Function points or COCOMO
 - Expert estimations/judgement
 - Estimation by analogy
 - Top-down estimation
 - Bottom-up estimation

Name/Nimi/Namn: _____

Student number /Opiskelijanumero /Studentnummer: _____