NTNU – Trondheim
Norwegian University of
Science and Technology

Department of Computer and Information Sciences

Examination paper for TDT 4242
Advanced Software Engineering

Academic contact during examination: John Krogstie

Phone: 93417551

Examination date: June 2. 2016

Examination time: 09:00 AM to 1:00 PM

Permitted examination support material: Code C – Pocket calculators allowed - In addition it is allowed to bring a print-out of the following standards:

- ISO 9126 Part 1 Quality characteristics and sub-characteristics
- IEEE 829-1988 IEEE Standard for Software Test Documentation
- IEEE 1059-1993 IEEE Guide for Software Verification and Validation Plans

Other information: Exam developed by John Krogstie and checked by Hallvard Trætteberg

The points on the sub-questions will add up to 100%.

The exam-text is below found both in English and norsk bokmål. The English version is to be regarded as the original if there are discrepancies. If you consider the information given *incomplete* or *inaccurate*, then make a note of the assumptions you find it necessary to make in order to answer the question. Good luck!

Language: English, Norsk bokmål

Number of pages: 6

Checked by:

_____

Date Signature

English version

In appendix 1, the IT activity of a public sector organization is described. This organization will be referred to in several of the tasks.

Task 1 – Requirements engineering (35 %)

Up till now, the requirements specification work for new and replacement systems in the organization described in Appendix 1 has been developed in a rather ad-hoc manner. However, due to some problems in the two latest projects, the company has decided to start using one or more predefined and documented processes for requirements specification. After some research, they are down to three candidates: User stories, Use cases and Goal Oriented Requirements Engineering (also known as GORE).

a) Give an overview of important quality characteristics of a *requirements specification* (10%)

SG: Two outsets for this: The one in SEQUAL article, i.e. the long list:

- Unambiguous
- Complete
- Correct
- Understandable
- Verifiable
- Internally Consistent
- Externally Consistent
- Achievable
- Concise
- Design-independent
- Traceable
- Modifiable
- Electronically Stored
- Executable/Interpretable/Prototypable
- Annotated by Relative Importance
- Annotated by Relative Stability
- Annotated by version
- Not Redundant
- At Right Level of Detail
- Precise
- Reusable
- Traced
- Organised
- Cross-refernced

A subset of these is listed with possible metrics:

Alternatively start out from the list provided in IEEE 1059-1993 (part of material included) : 5.5.3.2: Software requirements evaluation: *Evaluate SRS requirements for correctness, consistency, completeness, accuracy, readability, and testability. Assess how well SRS satisÞes software system objectives. Assess the criticality of requirements to identify key performance or critical areas of software: Can also touch on traceability and relation to other systems:*

Not all aspects (of the long list) needs to be mentioned. Important that it is the quality of the RE and not the system that should be mentioned. Extra positive if these are used in the argumentation in the following questions

b) Describe the strong and weak points of *User stories*, *Use cases* and *GORE* for requirements specification and management from the point of view of the organization described in Appendix 1 (15%)

Must make some assumption on the organization. Trouble with requirements management, large distributed org. Both replacements and some new development. Users with diverse backgrounds. Focus also e.g. on securiy

User stories

Strong ponits

- ***Easy to communicate to the customers and use as a basis for discussions***
- ***Easy to understand***
- ***Quick to make***
- ***Manageable content***
- ***Open for change***

Weak points

- ***Give little support for overall structuring of requirements relative to the system***
- ***Not very precise***
- ***Primarily the functional requirements***

Use case

• ***Strong points:*** ○

Easy to communicate to the customers and use as a basis for discussions
The use case diagrams are easy to understand
The textual use cases can be written so as to be close to the users' process
Both used case diagrams and textual use cases can be developed in a stepwise manner as our understanding of the problems improve

• ***Weak points:***

When we use textual use cases it is hard to get an overview of the full functionality
Advanced and useful concepts such as <<include>> and <<extend>> are difficult to understand for most users
Only functional requirements

GORE:

• ***Strong points:*** ○

Help us to focus on "what to achieve" instead of "how to achieve it"

Has useful patters such as "achieve", "prevent" and "maintain" which help us to organize the elicitation process

Have clear rules that can be used to decide when we have goals, sub-goals, assumptions and concrete requirements.

Can represent both functional and non-functional requirements

• Weak points:

Can be difficult for users to think about goals – they think about jobs or missions.

GORE is more abstract than use cases, thus easily putting the analyst in the driver's seat, running the risk of leaving the user behind

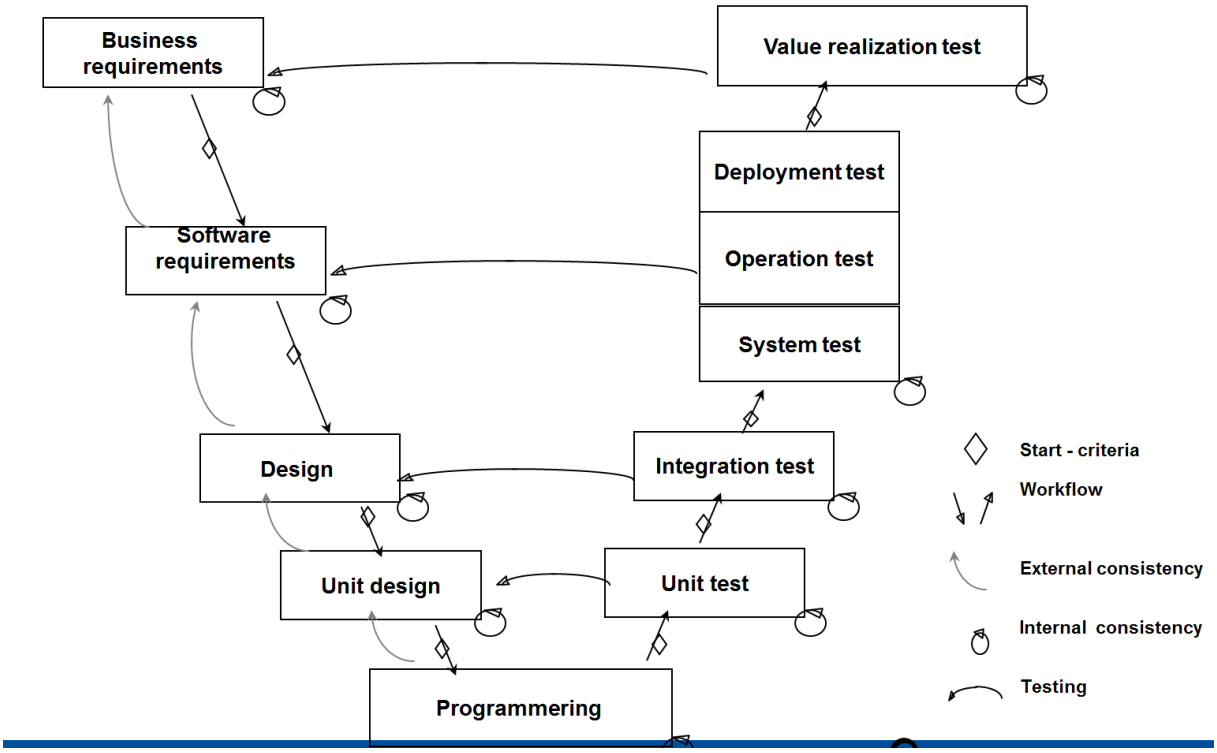
c) If you are to select only *one* of the three methods discussed in this task for the organization, which would you recommend and why? (10%)

SG: Since there are different development situations, ideally one would use a combination of techniques. Here I would aim for use cases with textual descriptions. Give some overview and structure, but not too rigid and system-oriented, thus should be easy to communicate. As with user-stories you can structure the project according to addressing them

Task 2 – Testing (30%)

The V-model connects testing activities and development activities in a software development project. This model also holds for agile projects, which can be considered as a sequence of waterfall projects, each with its own V-model.

a) Describe main *development and testing activities* of the V-model (8%)



The value realization test is often called acceptance test. (OK to not include the business level)

b) Select a *test approach* for each testing activity in the V-model (Grey box testing is an example of a test approach). Give a short description of each test approach, and explain why you have chosen it. (8%)

SG:

- Unit test: White box testing
- Integration test: White box or grey box test
- System test: Black box test
- Acceptance test: Black box test

c) Describe TDD – *Test-driven development* (9 %)

SG : Develop tests first and then the code to pass the test. Originally an agile menhod, but do not have to be. Some advantages:

- Writing clear (testable) requirements
- Development in small steps. This will make debugging easier since we will have small code chunks to debug.
- Minimalistic code and enforce the YAGNI principle – "You Ain't Gonna Need It"

d) Discuss briefly to what extend you would regard TDD to be an appropriate method for the organization described in Appendix 1 (5%)

SG: TDD primarily for greenfield development, on which there are few in the organization. It can be combined with user stories and use cases though, thus one can reflect on this combination. Thus it might not be a good idea to include this (can use an adapted version with legacy code

Task 3 – Non-functional requirements (15%)

a) What are *security requirements*? Give examples of *at least three types* of security requirements (5%)

SG: Security requirements has to do with how to avoid persons to harm the system. IS= 9126:

> The capability of the software to prevent unintended access and resist deliberate attacks intended to gain unauthorised access to confidential information, or to make unauthorised modifications to information or to the program so as to provide the attacker with some advantage or so as to deny service to legitimate users.

Some examples. Authentification, authorization, privacy, non-repudiation

b) *Mis-use cases* can be used to represent security requirements. Describe the technique and the relationship between mis-use cases and use cases. (10%)

SG:

- Aims to Identify possible misuse scenarios of the system
- The basic concept describes the steps of performing a malicious act against a system and ways to mitigate this
- The process is the same as you would describe an act that the system is supposed to perform in a use case

Task 4 – Bimodal IT and Lean Startup (20%)

Returning to the case in Appendix 1, the organization has realized that due to the variety of systems in their portfolio, they need to use several approaches to system development in parallel. To follow up this, they are investigation an approach called bimodal IT.

a) What is *bimodal IT*? (5%)

   SG:

   The practice of managing two separate, coherent modes of IT delivery, one focused on stability and the other on agility.

   a. Mode 1 is traditional and sequential, emphasizing safety and accuracy.
   b. Mode 2 is exploratory and nonlinear, emphasizing agility and speed

b) To support the different modes in bimodal IT, different requirements specification techniques might be applicable. Which of the three different approaches mentioned in task 1 (User stories, use cases and GORE) do you think best fit the different modes of bimodal IT? (5%)

   Mode 1: GORE: Mode 2: Uses stories

c) *Lean Startup* can be seen as an alternative to traditional agile development for applications in new domains. Describe the Lean Startup technique (5%)

SG: Lean startup is an technique where you focus on quickly coming up with least viable products because of uncertainty: (from sides: A human institution to create new products and services under conditions of extreme uncertainty.Build-measure-learn-cycle It differs from traditional agile development across a number of areas:

	Agile	Lean Startup
Cross functional teams	✔	✔
Retrospectives	✔	✔
Pair-programming and technical workshops	✔	✔
Deployment to production on first day	✖	✔
Tasks derived from user stories	✔	✖
Tasks derived from experiments	✖	✔
Estimates used to measure progress	✔	✖
Validated customer feedback used to measure progress	✖	✔
The back-log tells us what to do next	✔	✖
Validated learning tells us what to do next	✖	✔
Aim for maximum test coverage	✔	✖
Aim for optimal statistics about user behavior	✖	✔

d) For what kind of tasks (if any) in the case organization described in Appendix 1 would you recommend to use Lean Startup? (5%)

SG: It might be applicable to the IoT-applications. On the other hand, since such applications might influence the communication infrastructure (wireless network) one might also want to control this a bit, especially before deploying new solutions.

Norsk bokmål

I vedlegg 1 beskrives IT-aktiviteten til en virksomhet innen offentlige sektor. Virksomheten vil bli referert til i flere av oppgavene under.

Oppgave 1 – Kravspesifikasjon (35 %)

Så langt har arbeidet med kravspesifikasjon for nye systemer og erstatningssystemer i virksomheten beskrevet i vedlegg 1 blitt gjort på en ad-hoc måte. På de siste par prosjektene har man hatt litt problemer grunnet dette, og man har bestemt seg for å begynne å bruke en eller flere dokumenterte metoder for kravspesifikasjon. Etter litt undersøkelser har man identifisert 3 kandidater: User stories, use cases og Goal Oriented Requirements Engineering, også kjent som GORE.

a) Gi en oversikt over viktig kvalitetskarakteristika for *kravspesifikasjoner* (10%)
b) Beskrive sterke og svake sider ved *User Stories*, *Use Cases* og *GORE* for kravspesifikasjon sett i lys av situasjonen i organisasjonen.
c) Hvis du skal velge *bare en* av disse tre metodene for virksomheten, hvilken ville du anbefale og hvorfor? (10%)

Oppgave 2 – Testing (30%)

V-modellen knytter utviklingsaktiviteter og testaktiviteter i et systemutviklingsprosjekt. Dette er også en modell som gjelder for smidige prosjekt, som kan ses på som en sekvens av fossefallsprosjekt med hver sin V-modell.

a) Beskriv de sentrale *utviklingsaktivitetene og testaktivitetene i V-modellen* (8%)
b) Velg en *testmetode* for hver test-aktivitet i V-modellen. Grey-box testing er et eksempel på en test-metode). Gi en kort beskrivelse av hver testmetode og forklar hvorfor du har valgt denne. (8%)
c) Beskriv TDD – Test-driven development (*test-drevet utvikling*) (9 %)
d) Diskuter kort i hvilken grad du anser TDD for å være en egnet metode for organisasjonen beskrevet i vedlegg 1 (5%)

Oppgave 3 – Ikke-funksjonelle krav (15%)

a) Hva er *sikkerhetskrav*? Gi eksempler på *minst tre ulike typer* sikkerhetskrav (5%)
b) *Mis-use cases* kan brukes for å representere sikkerhetskrav. Beskriv denne teknikken og sammenhengen mellom mis-use cases og use cases.(10%)

Oppgave 4 – Bimodal IT og Lean Startup (20%)

Virksomheten beskrevet i vedlegg 1 har innsett at grunnet den store variasjonen de har i systemporteføljen må de bruke ulike systemutviklingsmetoder for ulike systemer parallelt. Før å følge opp dette ser de på en metode som kalles bimodal IT.

a) Hva er *bimodal IT*? (5%)
b) For å støtte ulike modi i bimodal IT er det ulike kravspesifikasjonsmetoder som er best egnet. Hvilke av de tre kravspesifikasjonsteknikkene beskrevet i oppgave 1 (User stories, use cases and GORE) er best egnet for de ulike modiene i bimodal IT? (5%)
c) *Lean Startup* er en alternativ teknikk til tradisjonell smidig utvikling for å utforske systemutvikling på nye områder. Beskriv Lean Startup-teknikken (5%)
d) For hvilke aktiviteter (om noen) i virksomheten beskrevet i vedlegg 1 vil du anbefale bruk av Lean Startup ? (5%)

Appendix 1 – Case organization

The organization is in the public sector and has 7000 employees in three different cities. 250 of these work with IT, Half of these work in a centralized IT department while the rest work with IT in other departments. Of the total time spent on IT-related work the organization spends 15% on development and 20% on maintenance. The rest is used on operations (approximately 50%) and user support. Application portfolio upkeep is 57%, i.e. more than 40% of development and maintenance time is used to provide new functionality for the users of the organization.

The IT department runs much of the systems themselves, but they also outsource some activities, in particular operations. In the future they plan to also outsource more of the development and maintenance of applications to be operated by external parties using cloud technology solutions.

The organization has two different reasons for developing replacement systems. Because they are a public organization they are required to put existing system out to tender from time to time. This often results in replacing the system with a new one. The other reason for replacing a system is that the underlying technology is outdated. When they are replacing a system they also add new functionality, starting with developing an updated requirement specification.

The IT director explains that they have a comprehensive regime associated with maintenance. They use a methodology called ITIL to handle maintenance requests. It can evaluate how important the change is, and can reduce time spent on error correction. They also have some individuals that work only with security since the systems manage data about persons. This data should not be accessed by the wrong person or be tampered with.

A big organization like this has a lot of IT challenges. Different users have very different IT-needs, and varying IT skills. They see that the future can include the need to integrate a lot of sensors and actuators in solutions utilizing the internet of things, which puts pressure on the wireless network.

They also experience that there are more and more interactions between individual systems, with many applications that need to work together, so it is essential to support a common integration platform. The organization also gets a bigger and bigger application portfolio, which requires good portfolio management to be able to prioritize request for changes across the different applications.

Another aspect they see is that the development in the industry is happening fast. To be able to follow these trends the organization need to be able to act quickly in new business and technology areas and at the same time keep the operation of the core system stable.