NTNU – Trondheim
Norwegian University of
Science and Technology

,

Department of Computer Sciences

Examination paper for TDT 4242 Advanced Software Engineering

Academic contact during examination: Jingyue Li

Phone: 9189 7446

Examination date: May 26. 2017

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

Permitted examination support material: Code C – Pocket calculators allowed

Other information: Exam developed by Jingyue Li and checked by John Krogstie

Language: English

Number of pages: 7 (including this page and the Appendix 1)

Checked by:

Date Signature

Introduction

In this exam, you can score a maximum of 50 points. The remaining 50 points for the semester comes from the compulsory exercises.

If you feel that any of the problems require information that you do not find in the text, then you should

• Document the necessary assumptions

• Explain why you need them

Your answers should be brief and to the point.

Task 1 – Requirements engineering (14 points)

The case is a snow clearing robot. See Appendix 1 for more details

Task 1.1 Requirement quality (6 points)

Try to categorize the textual requirements in Appendix 1 according to the requirements quality metrics: *ambiguity*, *inconsistency*, *forward referencing*, *opacity*, and *noise*. (4 points). If one requirement has several quality issues, list and explain all of them.

Then, try to fix the requirements quality issues of each requirement in the snow clearing robot in Appendix 1 and write down the improved requirements. (2 points)
(Note: There is no single correct answer of improved requirements of the snow clearing robot. You can just formulate improved requirements based on what you think a snow clearing robot should be. The key point here is to show you know how to create high quality requirements).

SG (There are possibly several quality issues in each requirement. The solution below are the minimum requirements quality issues that should be identified and corrected.):

1. The robot is switched on and off with a button on the instrument panel.

   (Ambiguous, says a button, not which button)

2. When the robot identifies that the snow on the ground around itself reaches to a certain depth centimeter, the robot should automatically start and move to clear the snow.

   (forward referencing, says a certain depth, does not specify the number)

3. When the user presses the remote-control key, the user can take over and have a full control of the robot to be able to move it forward, backward, left turn, and right turn. However, the robot can still make its own decision to move or not

   (Inconsistency)

4. When the air temperature is below 0 degree Celsius, the robot should go to the charge site to charge its battery.

(opacity, no clear link between air temperature and battery charge)

5. The robot should be able to identify obstacles and avoid collision with the obstacles.

   (Ambiguity, which type of obstacles, dogs, human, cars?)

6. The robot should be able to turn to a safe mode when it is in an unsafe environment.

   (forward referencing, does not define what a safe mode is and what an unsafe environment mean)

7. The robot should maintain a safe speed when clearing the snow.
   (forward referencing, does not define the value of safe speed)

8. The robot should be able to blink light to another snow clearing robot to say hello when they meet.
   (*noise. It is not relevant requirement*)

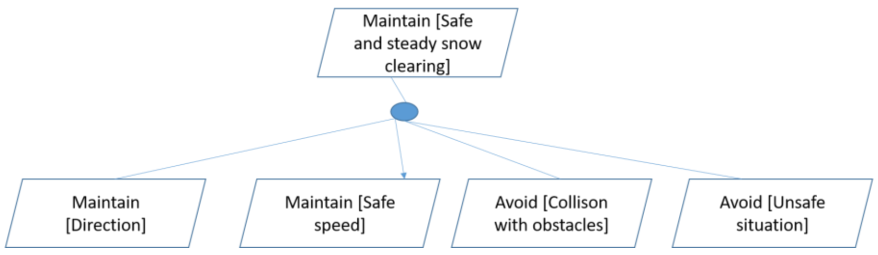
Task 1.2 GORE (4 points)

Based on your improved requirements, use the informal temporal patterns below to describe two upper levels of goal decomposition for the snow clearing robot case in Appendix 1.

Achieve [TargetCondition]
Cease[TargetCondition]
Maintain[GoodCondition]
Avoid[BadCondition]

Improve[TargetCondition]
Increase[TargetQuantity]
Reduce[TargetQuantity]
Maximise[ObjectiveFunction]
Minimise[ObjectiveFunction]

SG (one out of many possible solutions):


Maintain [Safe and steady snow clearing]
Maintain [Direction]
Maintain [Safe speed]
Avoid [Collison with obstacles]
Avoid [Unsafe situation]

Task 1.3 Requirement prioritization (4 points)

Prioritize the following high-level functional requirements of an **Intelligent Traffic Control System (ITCS)** using the cumulative voting method (2 points) and binary priority list method (2 points).
(Note: There is no single correct answer of ranking of the requirements. The key point is to show that you know how to use the two methods to prioritize requirements)

Req1. The ITCS system should be able to log and store traffic data collected through sensors into a server.

Req2: The ITCS system should be able to manage the traffic light to prioritize emergency vehicle.

Req3: The ITCS system should be able to manage the traffic light to decrease the amount of traffic congestions.

Req4: The ITCS system should be able to manage the traffic light to minimize the pollution produced by road traffic.

Req5: The ITCS system should be able to manage the traffic light to minimize pedestrians' waiting time when passing crosses.

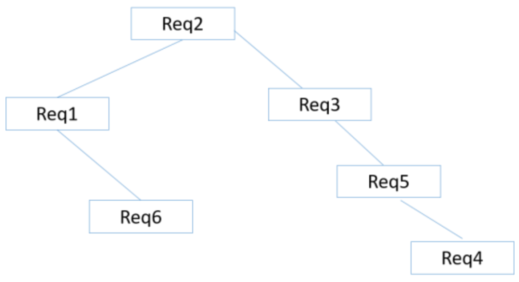
Req6: The ITCS system should be able to analyze the traffic data and make a report.

SG (one out of many possible solutions):

- *Cumulative voting method*

  Req1(8%), Req2(35%), Req3(25%), Req4(12%), Req5(15%), Req6(5%)

- *Binary priority list method*



Task 2 – Testing (15 points)

Task 2.1 Domain testing (3 points)

An online shopping web application has three variables (Availability, Payment method, and Delivery method). The possible values of the variables are as follows.

- Availability: Available (AVA), Not In Stock (NIS), DIScontinued (DIS)
- Payment method: Credit Card (CC), Gift Voucher (GV)
- Delivery method: Mail (MA), UPS (UPS), Fedex (FE)

You task is to write all the 2-way combinatorial test cases (Abbreviations of each variable value can be used)

SG:
If 2-way combinatorial must cover the following pairs (If all pairs are listed correctly, 2 points will be granted)
Availability-Payment method: AVA-CC, AVA-GV, NIS-CC, NIS-GV, DIS-CC, DIS-GV
Availability-Delivery method: AVA-MA, AVA-UPS, AVA-FE, NIS-MA, NIS-UPS NIS-FE, DIS-MA, DIS-UPS, DIS-FE
Payment method- Delivery method: CC-MA; CC-UPS; CC-FE, GV-MA, GV-UPS, GV-FE

9 test cases are needed to cover all the pairs (If the test cases are listed correctly, all 3 points will be granted)

Availability	Delivery	Payment
AVA	MA	CC
AVA	UPS	GV
AVA	FE	CC
NIS	MA	GV
NIS	UPS	CC
NIS	FE	GV
DIS	MA	CC
DIS	UPS	GV
DIS	FE	CC

Task 2.2 Regression testing (6 points)

Suppose we have a code snippet as follows. We initially have a test set T with 4 test cases to test the code. The 4 test cases in the test set T are:

t1: <x = 1, y = 2>
t2 : <x = 1, y = 3>
t3 : <x = 3, y = 1>
t4: < x= 2, y = 2>

If the line “return b*b” in function f1 is changed to “return a*b”, your task is to choose “safe re-testable subset” from T using graph-walk approach with CFG (Control Flow Graph) and explain in details how you choose the subset.

```
int  M (int x, int y){
  int z;
  if (x>=y)
     z = f1 (x, y);
  else
     z = f2(x, y);
  return z;
}
```

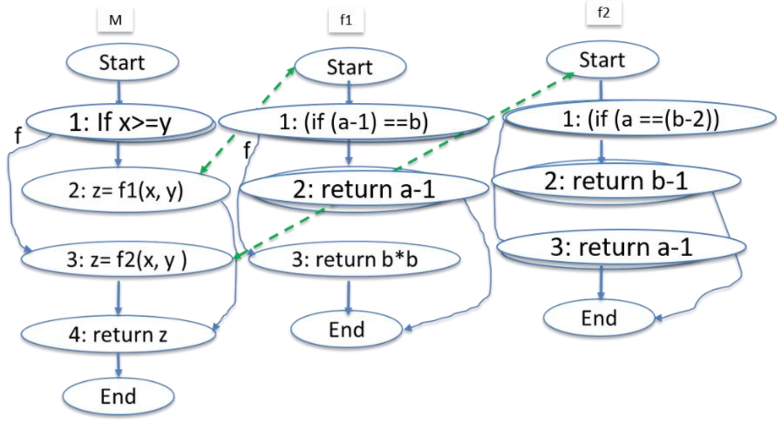
```
int f1( int a, int b){
  if ((a-1) == b)
     return a-1;
 else
     return b*b;
}
```

```
int f2( int a, int b){
  if (a == (b-2))
     return b-1;
 else
     return a-1;
}
```

Code snippet

SG:

- Step 1: Establish trace between test case and CFG (control flow graph) nodes



Test set T
t1: <x = 1, y = 2>
t2 : <x = 1, y = 3>
t3 : <x = 3, y = 1>
t4: < x= 2, y = 2>

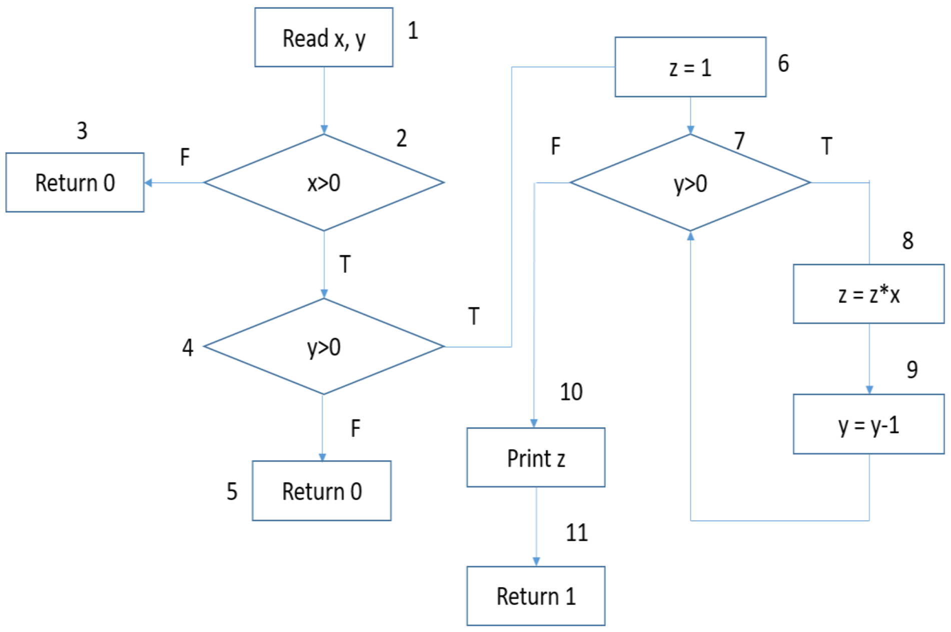
Functions	**Nodes**			
	1	2	3	4
M	t1, t2, t3 t4	t3, t4	t1, t2	t1, t2, t3, t4
f1	t3, t4	-	t3, t4	-
f2	t1, t2	t2	t1	-

- Step 2: Compare P and P' to find differences
  Node 3 of f1 has been changed.

- Step 3: Select test cases from T that traverse the changed CFG nodes
  Test cases t3 and t4 will be chosen for regression testing, because they traverse the changed CFG node

Task 2.3 Dataflow testing and test coverage (6 points)

For the following CFG (Control Flow Graph), find out paths to be covered for variables x, y, z, if All Uses (AU) is the test strategy to be applied.



SG (The paths must cover the following paths, although the paths can be much longer ones):
(1, 2, x)
(1, 8, x)
(1, 4, y)
(1, 7, y)
(1, 9, y)
(9, 7, y)
(9, 9, y)
(6, 8, z)
(6, 10, z)
(8, 8, z)
(8, 10, z)

Task 3 – Advanced topics (15 points)

Task 3.1 (3 points)

Explain the three high level categories of OSS license types and give one example license in each category.

SG:

- *Strong-copyleft (e.g. General Public License (GPL))*
  - *Once software is licensed by a developer, the* ***derivative*** *work (includes major copyright-protected elements of an original) must be licensed similarly*
- *Weak-copyleft (e.g. Lesser General Public License (LGPL))*
  - *Allows developers integrate software released under the LGPL into their own (even proprietary) software without being required to release the source code of their own components*
  - *Primarily used for libraries*
- *Non-copyleft (e.g. Berkeley Software Definition (BSD) and MIT license)*
  - *No obligation to inherit the original license*

Task 3.2 (3 points)

Explain what “operating system-centric” software ecosystem is (1 point) and list its success factors (1 point) and challenges (1 point).

SG:

Characteristics

- *Domain independent*
- *Installed for every device*
- *Focused on stand-alone applications*
- *Focus on development tools for developers*

Success factors

- *Number and relevance of applications built*
- *OS needs to evolve constantly*
- *Number of customers*

Challenges

- *Variation in underlying hardware configurations*

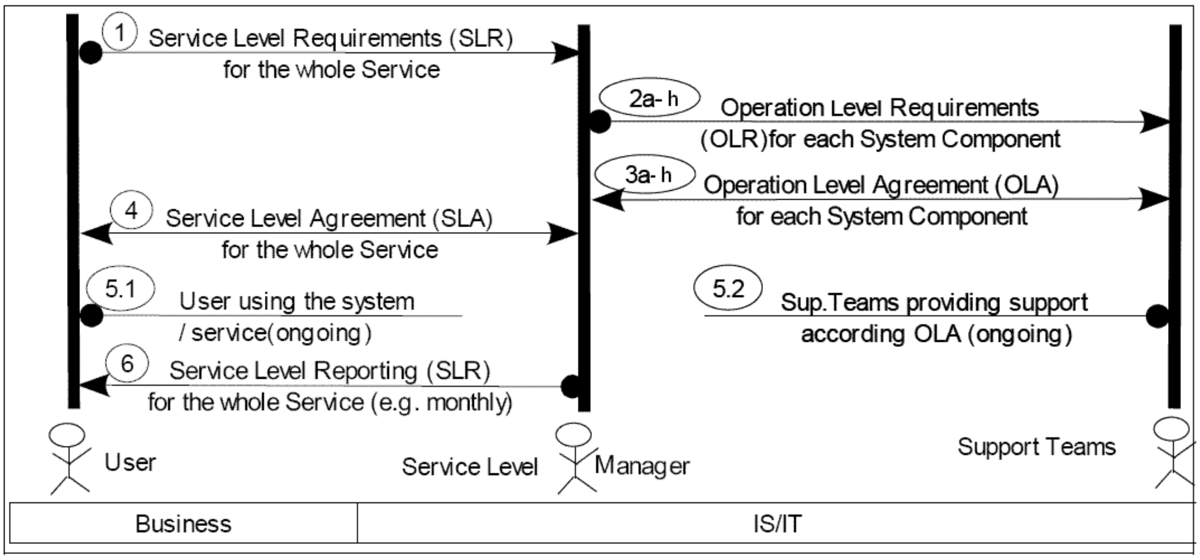
Task 3.3 (4 points)

Explain SLA (Service level agreement) and its content (1 points), OLA (Operations level agreement) and its content (1 points), and the relationship between SLA and OLA (2 points).

SG:

- ***SLA*** *is the overall agreement between IS/IT and the business department, usually includes*
  - *Services being made available*
  - *Service performance*
  - *Cost to provide the service*
  - *Service provider and customer responsibility*

  - *Performance monitoring and report*
  - *Disaster recovery process*
  - *Periodic review process*
- ***OLAs** are agreements between different IS/IT-departments and the Service Level Manager*
- *Relationship between SLA and OLA (A brief explanation of the idea of the chart is ok)*


1 Service Level Requirements (SLR) for the whole Service
2a-h Operation Level Requirements (OLR)for each System Component
3a-h Operation Level Agreement (OLA) for each System Component
4 Service Level Agreement (SLA) for the whole Service
5.1 User using the system / service(ongoing)
5.2 Sup.Teams providing support according OLA (ongoing)
6 Service Level Reporting (SLR) for the whole Service (e.g. monthly)
User
Service Level Manager
Support Teams
Business
IS/IT

Task 3.4 (5 points)

Explain why systematic reading techniques outperforms unsystematic reading techniques (4 points), and list and explain one systematic reading technique (1 point).

SG:
Unsystematic reading, where one or more readers scan through the document and look for defects, which are shaded as vertical, horizontal, or slanted lines. There are at least two problems with that approach owning to its unsystematic nature: (a) there are overlaps on the portions covered by different readers, and (b) there are regions that are not covered by any reader.

Systematic reading, where different readers are purposely selected to exam the software artifact, based on their individual expertise. They are also given specific instructions on where and how to detect defects. These two ensure that there will be no or minimal overlaps among what they will cover and the entire document is covered. Systematic approaches with specific responsibilities improve coordination and reduce gaps, which increases the overall defect detection effectiveness of review.

Systematic reading techniques: Defect-based reading, perspective based reading.

Task 4 – Multiple choice (6 points, one point for each question. You get the one point if all correct answers are chosen)

1) What are the desirable properties of the component-based architecture?

a) Parameterizable components
b) Components should not be customizable
c) Supports component development in multiple programming languages
d) Allows easy distribution of components from seller to buyer

SG: a, c, d

2) Which are the challenges of service oriented software engineering?
   a) Dynamic software evolution
   b) Single point of failure
   c) Complex negotiation process
   d) Hard to understand when automated

SG: b, c, d

3) What are benefits of TDD (Test-Driven Development)?
   a) TDD is always applicable
   b) TDD can lead to higher code coverage
   c) TDD will guarantee a shorter development time
   d) TDD helps simplify debugging

SG: b, d

4) What are differences between Model-Based Testing (MBT) and traditional manual testing?
   a) For MBT, test cases are tightly coupled to the model
   b) For MBT, test cases are generated automatically from models
   c) For MBT, there is a high cost in early phases of the project
   d) For MBT, test oracles still need to be input manually

SG: a, b, c

5) What can cause project cost/effort derives from its original cost/effort estimation?
   a) Pressure from managers to give low bid to get the project
   b) Over-optimistic estimates
   c) Frequent major requirement changes
   d) Major changes in design and implementation

SG: a, b, c, d

6) Which are advantages of scenario-based testing?
   a) Easy to reuse
   b) It is a user focused testing
   c) Does not require working features before testing
   d) Can expose requirement related issues

SG: b, d

Appendix 1 – Snow clearing robot

(Note: This picture is just to illustrate what a snow clearing robot is. This robot is not necessary linked to the requirements below)

Requirements:

Req1: The robot is switched on and off with a button on the instrument panel.

Req2: When the robot identifies that the snow on the ground around itself reaches a certain depth, the robot should automatically start and move to clear the snow.

Req3: When the user presses the remote-control key, the user can take over and have a full control of the robot to be able to move it forward, move it backward, make it turn left, and make it turn right. However, the robot can still make its own decision to move or not.

Req4: When the air temperature is below 0 degree Celsius, the robot should automatically go to the charge site to charge its battery.

Req5: The robot should be able to identify obstacles and avoid collision with the obstacles.

Req6: The robot should be able to turn to a safe mode when it is in an unsafe environment.

Req7: The robot should maintain a safe speed when clearing the snow.

Req8. The robot should be able to blink the light to another snow clearing robot to say hello when they meet.