DAT321/DIT847:
Software Quality

Welcome to the examination for the *Software Quality* course. The examination is intended to last for max **4 hours** and is intended to be **anonymous** (i.e., the teacher grading your exam will not know your name). Therefore, it is important that you follow the instructions (in the separate exam cover sheet) and **do NOT leave any information that would reveal your name.**

Each question has a number of points assigned shown in the square brackets. When the question is broken down into smaller sub-questions the part of the points for that specific sub-questions are also shown as following:

1. *[10 pts].*
    a. *[2 pts]*
    b. *[8 pts]*

The percentage of points and the corresponding grade is presented below (100 points in total):

% of points	DAT321	DIT847
[0, 50%)	U	U
[50%, 65%)	3	G
[65%, 85%)	4	G
[85%, 100%]	5	VG

Write your answers in the exam sheets. Before handing in your exam, number and sort the sheets in task order. Write your **anonymous code** and page number on every page!

It is important that you write **clearly** so that the examiner can read you. If your handwriting is unreadable, then you will not get any points for that question. We will NOT assess grammar or spelling as long as your answer is readable, understandable and unambiguous.

The questions in this exam refer to the **ISO 25010:2011** that categorises internal and external software quality attributes into eight characteristics.

Simple calculators are also allowed, but NOT calculators in mobile phones.

Questions about the exam contact:
Richard Torkar, tel. 031 772 57 19, richard.torkar@cse.gu.se
Francisco Gomes, tel. 031 772 69 51, francisco.gomes@cse.gu.se

The exam review is scheduled for 2019-11-29, between 13:30 – 15:30 at Jupiter building, 4th floor, Room 427.

The questions in this exam are related to the following context. Note that the answers should, in turn, be justified based on i) the theory and terminology from software quality and ii) their connections to the elements in this context (e.g., team, product, processes, architecture, etc.).

Context: You are hired by GPS-Co, a company responsible for developing a cloud-based system that offer various GPS services. You will be the quality assurance manager for teams B and C (see description of the teams below).

Software product:
The product is composed of several services, all of them related to GPS positioning and navigation. The product uses a microservices architecture, where the different services are loosely coupled and independently deployable. The services are accessed and integrated via APIs endpoints (i.e., entry points that work as method calls). The company requires all teams to use Continuous Integration practices and tools.

Customers, business, teams and software development process:
Currently GPS-Co has contracts with other companies that use the services provided by your software product. Currently, your main customer is the FoodDelivery-Co, which delivers food orders from supermarkets to customers. FoodDelivery-Co uses the GPS services on their fleet of trucks that deliver all food orders from supermarkets to customers who ordered online. Truck drivers use a Graphical User Interface (GUI) in the truck's dashboard panel. The GUI sends requests to the GPS-Co API, such as suggestion of routes that include all delivery points.

Today, GPS-Co signed a contract with another customer, Automotive-Co which is an automotive company that wants to use the GPS services for tracking and navigation of its manufactured cars. Therefore, as part of the contract, Automotive-Co will request several new features from your software product.

Team A is responsible for developing and maintaining the main features of your software product: i) the main application with the various GPS features, ii) the cloud infrastructure and platform where the service is hosted, iii) all non-functional aspects of the product. Team A has 50 members and is internally divided into small groups based on the specific services provided by the software product. Team A is composed of experience engineers with various testing and development skills.

Team B handles the development of customised features for FoodDelivery-Co, and is a small team with 8 people, all of them with wide experience in software development and agile software development. Team B reuses the features created by Team A in order to customize the product for FoodDelivery-Co. Since FoodDelivery-Co users rely on a GUI, Team A has one senior user experience (UX) designer.

GPS-Co is also hiring developers and testers for Team C that will work similarly to Team B but for Automotive-Co. Therefore, you must help with recruitment for this team.

1. **[20 pts]** Using the context above, answer the following questions:
   a. [15 pts] Choose three software product quality's characteristics, **and** provide examples of why they are relevant to the software product at GPS-Co.

Any of the eight below could work. Other examples could be provided, but it is important to clearly state the relevance.

Usability: They are relevant to the product from the end-user's perspective. Particularly, drivers from the truck fleet or future owners of cars from Automotive-Co, will use the GPS for navigation or positioning information.

Maintainability: They are relevant to the product because modification and change requests are common in any software project, hence software should be easy to maintain. For instance, Teams B and C will need to implement and/or request features to Team A such that maintaining the parts of the product should not compromise development and functioning of the product.

Reliability: They are relevant to the product because the product is available for drivers on the road, and other users that require information live (e.g., for the navigation functionality). For drivers, poor reliability can compromise safety since two of the customers use the product for driving.

Security: They are relevant because the product stores and processes personal information (e.g., current location of vehicles, or users). Poor security can lead even to safety and privacy issues with the users of the product.

Compatibility: They are specially relevant because our product is based on API, such that we microservices (and external clients of our application, such as the trucks), must be able to share information (e.g., navigation and positioning). For instance, in case we want our product to talk with other APIs to offer, for instance, weather forecast of specific places, compatibility becomes an important quality characteristic.

Portability: They are relevant if we consider that the product is accessed by hardware devices such as mobile phones from users, or even embedded in the trucks or cars of our customers. Therefore, the product must be able to work for various versions of hardware platforms.

Functional suitability: They are relevant because the product must meet the stated and implied needs of the users, such as the requests for customized features from our customers, or even the basics features expected from a GPS software system (e.g., navigation and positioning).

Performance efficiency: They are very relevant because the product runs on a cloud infrastructure that requires computational resources which affect the usage and functioning of our product. Particularly, GPS features are highly connected to information updated live (real-time), hence latency and limited capacity of resources can significantly affect our product.

a. [5 pts] Explain the differences between product quality and quality in use.

Product quality is connected to the quality of the product based on eight characteristics pertaining the product without considering the user or a specific context of usage. In turn, quality in use is the combined effect for the **user** of the eight software product quality characteristics, such that the software is being evaluated under a defined context that include the user.

b. [10 pts] Provide two examples of how to measure quality in use of your product. Your example must include activity, entity under investigation and viewpoint.

2. **[10 pts]** Using your knowledge on software quality measures, answer the following:
   a. [5 pts] McCabe and Halstead are two distinct complexity measures used in software quality. Are they used for internal or external quality? Justify your answer.

[1pts] Internal Quality.
[4pts] Because they are obtained from artefacts of the software product and relate to the static aspects of the software.

   b. [5 pts] What are the differences between both measures in terms of software complexity?

The differences are that…

[2pts] McCabe measures complexity based on the use of control flow constructs of the program, such as if, loops, for, etc.

[1pts] McCabe is connected to the testability of the software being analysed, or connected to the difficulties in testing/debugging the software.

[2pts] Halstead is based on operands and operators of the program, hence it is connected to the complexity of, e.g., understanding several variables and their corresponding values.

3. **[25 pts]** Using your knowledge on Verification and Validation (V&V) and exploratory testing, answer the following questions:
   a. [5 pts] Explain the differences between fault, failure and error.

   They differ in their location and meaning in software testing.
   [1pt] Mistakes or errors are made by humans when writing code, hence
   [2pt] Introducing a fault which is a wrong implementation located in the code.
   [2pt] When executing the code, if the fault is triggered, the software will show an unexpected behaviour, also referred as failure.

   [1pt] Alternatively, one can describe errors as the erroneous state reached after exercising the fault.

   b. [5 pts] Describe the advantages and disadvantages of using exploratory testing.

   (any selection of the below that accounts for max. 2-3 points)
   The advantages are that
   [1pt] stakeholders do not need to dedicate effort in writing or maintaining tests,
   [1pt] it is independent from the implementation since it is often done at the GUI level
   [1pt] it does not require knowledge of a specific programming language
   [1pt] it is at the same level as acceptance test
   [2pt] simultaneously, testers do three things: creates and runs tests, while learning about the SUT.

   (any selection of the below that accounts for max. 2-3 points)
   The disadvantages are:
   [2pt] It is very hard to automate, hence it is predominately manual.
   [2pts] Risk of humans becoming biased and testing only ideal/easy scenarios (“happy paths”).

   c. [5 pts] Considering the different levels of testing in the V-model, should exploratory testing be used in all levels of testing? Justify your answer.

      [1pt] No.
      [4pt] Because lower levels, such as unit and integration, requires complex testing activities that should be automated for continuous and repeatable checking. Exploratory testing is not entirely reproducible due to its exploratory nature. ET is mainly manual and driven by a human tester, which is not scalable at lower levels of testing.

d. [10 pts] A senior manager at GPS-Co asks whether the teams should use exploratory testing to improve test effectiveness at the company. Do you recommend exploratory testing to be used at GPS-Co? You must justify your answer based on the GPS-Co context provided such as teams, product, development process.

[2pt] Yes, we recommend.
[3pt] However, exploratory testing should be used in teams where products have access to a GUI (e.g., FoodTruck-Co).
[3pt] Since there are resources (e.g., the UX designer in Team A) used to build sensible GUI, testers could benefit from that UI through exploratory testing.
[2pt] However, since GPS-Co also has continuous integration, exploratory testing should not be used on lower levels of testing such as integration and unit, where the continuous integration pipeline requires automated tests.

4. **[15 pts]** GPS-Co wants to contribute with sustainability in ICT. You must *suggest* or *explain* two sustainable features of your software product. Your explanation must include: (i) a description of the feature, (ii) why is it connected to sustainability in ICT, and (iii) which sustainability dimension is it connected to.

Note: Here, you can interpret features as: new functionalities/services of your product, updates to the development process, or changes in the current software system.

A few examples (7.5 pts each example presented):

Example 1:

i) optimise performance efficiency to reduce energy consumption of the cloud infrastructure. ii) this connects to different dimensions depending on the perspective explained. For instance: the environment dimension for better energy efficiency; technical dimension for enabling long term usage and evolution of the product; economic if you consider the costs involved of using services in a cloud infrastructure (e.g., data traffic, Amazon Web Services, etc.).

Example 2:

i) Suggest optimised truck routes to FoodDelivery-Co make several deliveries within a specific time. ii) Again, could be more than one dimension: For instance: environment for less CO2 emission of trucks with more optimised route; economic if that could affect the business model with FoodDelivery-Co;

Example 3:

i) add feature for sharing routes for social activities (e.g., hitchhiking to work, tracking for hiking or biking). ii) connected to the social dimension since it allows to bring groups of people with the same interest;

Example 4:

i) add a user interface with support for voice recognition and text-to-speech, or other accessibility features. ii) connects to individual/social dimension since it is inclusive.

5. **[20 pts]** Determined to improve the test effectiveness, a senior manager at GPS-Co wants to compare two different test techniques (*Technique A* and *Technique B*) regarding their fault detection capability. Therefore, she asks you to setup an experiment where 50 participants will use each technique and record the number of faults revealed. She also divided participants in two groups: junior and senior engineers to verify if their experience affects the number of faults revealed. An example of the data collected during the experiment is shown below:

Faults	**Technique**	**Participant ID**	**Experience**
8	Technique A	P1	Senior
7	Technique B	P1	Senior
4	Technique A	P2	Junior
8	Technique B	P2	Junior
10	Technique A	P3	Junior
12	Technique B	P3	Junior
…			

Write down the *mathematical model definition* for this experiment using *any* variable names and priors of your choice. State the ontological and epistemological reasons for your likelihood. Remember to clearly state and justify the choices and assumptions regarding your model.

[1pt] In order to compare techniques, we must compare the prior distribution to see which technique is more likely to reveal more faults.
[1pt] We need a GLM (General Linear Model) to predict the number of faults.

[2pt] The number of faults can be affected by the choice of technique
[2pt] The number of faults can be affected by the experience/seniority of the tester.

[Lose between 5-10] Not a Poisson likelihood. Cannot be a Binomial, because a binomial needs a size, and cannot be normal because there are no negative faults. Other models will be analysed based on the justification provided.

[3pt] We choose a Poisson likelihood because fault is a positive number and tend to come with the rate of revealing faults (i.e., connected to running tests).
[2pt] Since we are using a Poisson likelihood, we need a link function, and that is a log function.

[3pt] The first predictor is the technique used to reveal faults because we expect that each technique has an effect on the number of faults revealed.

[3pt] The number of faults can also be affected by each participant individually, leading to more variance. Therefore, we need a varying intercept for each participant.

[5pt] The experience can be used as a predictor as well, but recall that they are connected to the participants. (Optional in case you did not choose varying intercepts)

[5 pts] The model below yields a posterior that allows us to predict number of faults based on different predictors:

$$faults \sim Poission(\lambda_i)$$
$$\log(\lambda_i) = \alpha + \beta_t * Technique + \beta_e * Experience + \alpha_{Subject[i]}$$
$$\alpha \sim Normal(0, 5)$$
$$\beta_t, \beta_e \sim Normal(0,1)$$

[3pts] Now, for the hyperparameters/priors of the varying intercepts:

$$\alpha_{Subject} \sim Normal(\mu_s, \sigma_s)$$
$$\mu_s \sim Normal(0,1)$$
$$\sigma_s \sim Cauchy(0,2)$$

[3pts] We choose an informative prior, since there is not prior knowledge about the data, and, particularly, we choose a flat prior.