Examination Assignment

Introduction to Automotive Software Development

# Agile Software Development

1. What is Scrum?

Scrum is a methodology that offers a framework for solving complex problems.

2. Name and briefly explain the pillars of Scrum.

* **Transparency**: everyone can see everything and knows where everybody else stands. Everyone is entitled to have a perspective.
* **Inspection**: it is the moment where we see if the results are actually being achieved, direct real feedback.
* **Adaption**: if the results are not as we wished, we adapt. Adaptation could happen to fix a problem, or to improve something that could be better.

[1][2];

3. Name and briefly explain the roles in a Scrum team

* **Product owner**: create the set of features, the “wish list”.
* **Scrum master**: secure that the project is progressing smoothly.
* **Developer team**: Build, test and use. Increment the spring.

[3]

4. Name and briefly explain the Scrum artifacts.

* **Product backlog**: Is the list of features that needs to be done to create or improve the product.
* **Sprint backlog**: It is a subset of the product backlog, big enough to make it into the spring goal.
* **Increment**: Is a step towards the product goal. They are additive and must be reviewed before marked as “done”.

[3]

5. Name and briefly explain the Scrum events.

* **Sprint**: It is where the ideas are turn into value. They are fixed time events, and consecutive.
* **Sprint planning**: Prepares the layout to be performed by the spring. This plan is created by the whole scrum team.
* **Daily scrum**: It is a 15 minutes meeting for the developers. The purpose is to inspect progress and adapt if necessary.
* **Sprint review**: The purpose is to inspect the outcome of the sprint.
* **Sprint retrospective**: The purpose of the Sprint retrospective is to plan ways to increase quality and effectiveness. "How can we do better next time?".

[3]

6. T-shirt Sizing is a Scrum story point estimation technique. Explain it.

T-shirt sizing is a tool for estimating developing times for a project.

It is done by assigning a t-shirt size to each task to represent the relative effort needed. The effort can be measured in time, complexity, scope, etc. It helps the leaders to recheck the team capacity. This technique is very informal and can be used quickly with large amount of items.

[4][5]

# Functional Safety

1. What is functional safety?

It is a general approach to improve the safety of electronic devices in modern cars.

2. Briefly explain the V-model used by ISO 26262 to develop products at software level

The V-model represents the life cycle of a development process. It highlights the relationship between test specifications at each stage of the process.

[6]

3. Explain the following ISO 26262 guidelines in software development level

a. Use of defensive programming. Why? What is defensive programming?

It is a style of programming intended to ensure the continuity of the functioning software. In this specific context, using defensive programming reduces the risk of software failure, which reduces chances of car accidents, which in turn increases the probability of saving lives in unfavorable situations.

b. Use of language subsets like MISRA C. Why? What is MISRA C?

MISRA C is a set of guidelines for the C language developed by the MISRA consortium. The aim of using MISRA C is to facilitate ​​safety, security, portability and reliability of the code.

[7]

c. Software unit testing. Why? What is unit testing?

Unit tes​​ting is a type of software where individual units or components are tested. The purpose is to validate that each unit of the software code performs as expected.

[8]

d. Software integration testing. Why? What is integration testing?

Software integration testing is the process of testing multiple units as a group. This test is focused on the interaction between the units.

[9]

# Automotive Software Architecture

1. What is software architecture and why do we need structured software?

The idea of software architecture is similar to general architecture in buildings. Certain buildings have special shapes depending on the functionality they need to perform. A hospital looks different than a restaurant, a bathroom looks and functions differently than a bedroom. Then applying the same idea, different software solutions use different architectures depending on the use. Examples: a bookkeeping system, a photo editing software, an automated car driving system, etc.

Structured software is a paradigm with the purpose of making the code more clear, easy to read and making the flow of control in a linear manner. In a structured program the whole solution is divided into smaller units called functions or procedures. These units are easy to test, easy to reuse, and easy to escalate into more complex solutions.

2. How can the architecture improve maintainability of a software?

The architecture offers the benefits of the design. In a structured paradigm the features can be separated into units or “functions”, then these units can be updated and tested individually. It is a type of atomization where all the procedures related to a specific topic are gathered into a minimized scope. There is no need to read the whole code to maintain a specific area.

3. What is AUTOSAR?

AUTOSAR refers to **AUT**omotive **O**pen **S**ystem **AR**chitecture. The name itself is auto descriptive. It is the special software architecture that ensures the best result for this specific environment. It defines different layers with specific tasks, and a long list of standards and guidelines.

[10]

4. What is the main idea behind the RTE layer in AUTOSAR?

RTE stands for Realtime Environment. The RTE layer works as an interface between the software components and the sensors of the vehicle. It gives real time information that should be processed and acted upon.

[10]

5. Explain the purposes of the following layers in the AUTOSAR basic software architecture

a. Microcontroller Abstraction Layer

The purpose of the MCAL is to directly access the peripheral modules and external devices that are mapped to memory, and with that make the software to be independent of the MCU.

[11]

b. ECU Abstraction Layer (ECUAL)

Offers uniform access to all functionalities of an ECU such as communication, memory or IO.

[12][13]

c. Services Layer

It is the higher layer of the Basic Software. Provides different types of background services such as network services, memory management and bus communication services for the application layer.

[12][13]

d. Complex Drivers

This layer spans widely from the hardware to the RTE layers. Everything that is not defined in the other layers fall into this layer. Provides the possibility to integrate special purpose functionality.

[13]

References:

[1] https://www.scrum.org/resources/blog/three-pillars-empiricism-scrum

[2] https://proagilist.es/blog/agilidad-y-gestion-agil/agile-scrum/los-tres-pilares-de-scrum/

[3] https://scrumguides.org/scrum-guide.html

[4] https://asana.com/resources/t-shirt-sizing

[5] https://www.c-sharpcorner.com/article/agile-story-point-estimation-techniques-t-shirt

[6] https://www.exida.com/Blog/whats\_a\_v\_model

[7] https://en.wikipedia.org/wiki/MISRA\_C

[8] https://www.guru99.com/unit-testing-guide.html

[9] https://www.guru99.com/integration-testing.html

[10] https://www.youtube.com/watch?v=H3dOaav3cx0

[11] https://www.renesas.com/us/en/application/automotive/autosar/microcontroller-abstraction-layer-mcal

[12] https://elearning.vector.com/mod/page/view.php?id=441

[13] ​​https://www.autosar.org/fileadmin/user\_upload/standards/classic/4-3/AUTOSAR\_EXP\_LayeredSoftwareArchitecture.pdf