Software Quality Assurance (WS20/21)
Problem Set 1

Problem 1: Embedded Systems

a) Please define the general term "system" according to Birolini and explicitly name the parts a system can encompass. Explain your answer in the view of aviation.

-

System: Technical and organizational means for the autonomous fulfillment of a task. Generally, a system can consist of hardware, software, people (service and maintenance personnel) and logistic assistance.

Example: Aviation

Hardware: Aircraft, tower, runway, ….
Software: Flight control software, software for in-flight entertainment system, …
People: Pilot, co-pilot, stewardess, ground personnel, …
Logistic assistance> Fueling, baggage handling, ticketing, ...

b) What is the difference to a "technical system"?

-

Technical system: System where influences by people and logistics are ignored

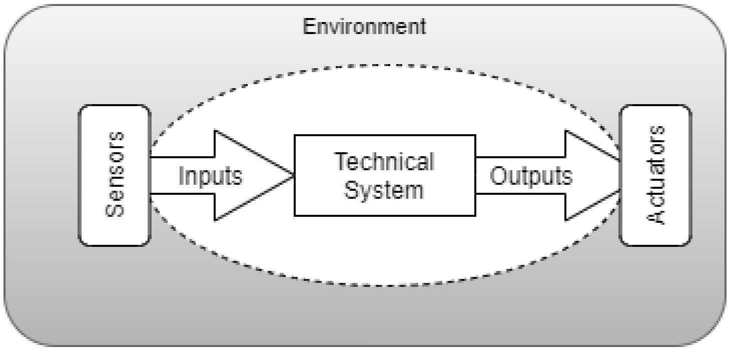
Example: Avionics

Hardware: Onboard flight control unit, ...
Software: Flight control software, …

c) For the analysis of a technical (embedded) system it is crucial to extract it from its environment. How can this be achieved? Please sketch your ideas.

-

The coupling between the technical system and its environment is modeled by "inputs" and "outputs" coming from "sensors" and going to "actuators".



d) Please list important non-functional requirements for embedded systems.

-

Real-time behavior: Deadlines must be met as required by the environment.
Minimal resource consumption: Memory, energy, dimensions, weight cost.
Dependability:

- Availability – Readiness for correct service.
- Reliability – Continuity of correct service.
- Safety – Absence of catastrophic consequences on users and environment.
- Confidentiality – Absence of unauthorized disclosure of information.
- Integrity – Absence of improper system state alterations.
- Maintainability – Ability to undergo repairs and modifications.

Problem 2: Reliability vs. Availability

Please explain the difference between “reliability” and “availability”.

-

Reliability: Measure for the ability of an item to remain functional, expressed by the probability that the required function is executed failure-free under given working conditions during a given time period.

Availability: Measure for the ability of an item to be functional at a given time.

Problem 3: Safety vs. Security

Please explain the terms “safety” and “security”. What is meant by “technical safety”? Please give examples for the safety of a failure-free system and the technical safety of a failure afflicted system.

-

Safety: State where the danger of personal or property damage is reduced to an acceptable value.

Security: Security is the concurrent existence of

a) availability for authorized users only
b) confidentiality
c) integrity

Technical safety: Measure for the ability of a failure afflicted item to endanger neither persons, property nor the environment.

Examples:

Safety of failure free system:

Dead man’s switch in a train
Two-switch starter of a molding press to prevent hand and arm injuries
Auto power-off of a lawn-mower if hands are taken away from handle

Technical safety of failure afflicted system:

Anti-lock braking system: proper pressure-release in brakes
Airbag: no ignition in non-accident situations

Problem 4: Failure, Fault

What is meant by the terms “failure” and “fault”? Please illustrate your answer by means of the “Ariane 5” disaster (see lecture).

-

- Fault, defect: Statically existent cause of a failure, (i.e., a bug). Usually the consequence of an error made by the programmer.
- Failure: Inconsistent behavior w.r.t specified behavior while running a system (happens dynamically during the execution) -> Each failure has a time stamp.
- Error: Basic cause for the fault (e.g., misunderstanding of a particular statement of the programming language)

Example: maiden flight of Ariane 5

Failure: Break-down of flight controller resulting in mechanical destruction of rocket.

Fault: Conversation of a 64-bit floating point variable into a signed integer with range -32768…..32767, leading to a data overflow within that variable

Error: Blind reuse of software components of Ariane 4, which have not been tested sufficiently within the new environment of Ariane 5

Problem 5: Hardware Failures vs. Software Failures

Please explain the differences between hardware failures and software failures.

-

Hardware failures	Software failures
Commonly caused by manufacturing errors or wear	It is commonly assumed that the software contains defects, which cannot be detected immediately
Traditionally, potential design faults within hardware are disregarded.	Software failures are a result of design errors that are contained in the product from the start and appear accidentally.
When the faulty component is substituted, its reliability becomes the initial values of this component	Occurrence heavily depends on operational profile
In the ideal case, the system contains no hardware defects in the beginning and therefore shows no hardware failures	After error correction the system reliability exceeds its initial value
The reliability of the system does not exceed its initial value through the substitution of components with new components	Faults that are introduced during debugging decrease reliability.

Problem 6: Correctness and Robustness

Please give your opinion on the following statements:

	True - False
Correctness has a binary character	True
Even if there are no defects, the program might not have to be correct	False
It can always be decided, whether an artifact is correct or not	False
An artifact is not consistent to its specification, if it is not correct	True
Robustness has a binary character	False
A correct system can have low robustness	True
Robustness is a property only of the implementation	False

Problem 7a: Quality Model

a) Quality characteristics might influence each other. Think about the following dependencies and figure out, whether the influences are positive or negative.

-

I. Safety – Availability: negative influence
   a. Safety – (-)-> Availability: Negative influence with stable safe state.
      Does not apply without stable safe state.

II. Availability – safety: cannot be defined

III. Safety – Reliability: cannot be defined. “Reliability has a positive influence”
   a. Reliability – (+)-> Safety: For safety relented services, reliability has a positive influence on safety.

IV. Availability – Reliability: cannot be defined.
   a. Reliability – (+) -> Availability: Tends to positive, but can be negative also.

V. Efficiency – Safety/Reliability: Efficiency only implies a negative influence when lack of resources.
   a. Efficiency – (-)-> Safety: Efficiency has a n`egative influence on safety
   b. Safety – (-)-> Efficiency: Introduction of safety related services can impair system efficiency.
   c. Efficiency – (-)-> Reliability: lacking resources at run-time might influence reliability.

b) Within ISO 9126 the following quality characteristics and sub-characteristics are defined. Please give a short definition for each one.

Quality characteristics	Sub-characteristics
Functionality	Suitability, Accuracy, Interoperability, Compliance, Security
Reliability	Maturity, Recoverability, Fault Tolerance
Usability	Learnability, Understandability, Operability
Efficiency	Time Behavior, Resource Behavior
Maintainability	Stability, Analyzability, Changeability, Testability
Portability	Installability, Replaceability, Adaptability, Conformance

-

Functionality: A set of attributes that bean on the existence of a set of functions and their specified properties. The functions are those that satisfy stated or implied needs.

a) Suitability: attribute of software that bears on the presence and appropriateness of a set of functions for specified tasks.

b) Accuracy: attributes of software that bear on the provision of right or agreed results of effects.

Interoperability: attributes of software that bear on its ability to interact with specified systems.

c) Compliance: Attributes of software that make the software adhere to application related standards or conventions or regulations in laws and similar prescriptions.

d) Security: attributes of software that bear on its ability to prevent unauthorized access, whether accidental or deliberate to programs and data.

Reliability: A set of attributes that bear on the capability of software to maintain its level of performance under stated conditions for a stated period of time.

a) Maturity: Attributes of software that bear on the frequency of failure by faults in the software.

b) Recoverability: attributes of software that bear on the capability to re-establish its level of performance and recover that data directly affected in case of a failure and on the time and effort needed for it.

c) Fault Tolerance: attributes of software that bear on its ability to maintain a specified level of performance in cases of software faults or of infringement of its specified interface.

Usability: A set of attributes that bear on the effort needed for use, and on the individual assessment of such use, by a stated or implied set of users.

a) Learnability: Attributes of software that bear on the users' effort for learning its applications.

b) Understandability: attributes of software that bear on the users' effort for recognizing the logical concept and its applicability.

c) Operability: Attributes of software that bear on the users' effort for operation and operation control.

Efficiency: A set of attributes that bear on the relationship between the level of performance of software and the amount of resources used =, under stated conditions.

a) Time Behavior: Attributes of software that bear on response and processing times and on throughout rates in performing its function.

b) Resource Behavior: Attributes of software that bear on the amount of resources used and the duration of such use in performing its function.

Maintainability: A set of attributes that bear on the effort needed to make specified modifications.

a) Stability: Attributes of software that bear on the risk of unexpected effect of modifications.

b) Analyzability: Attributes of software that bear on the effort needed for diagnosis of deficiencies or causes of failures, or for identification of parts to be modified.

c) Changeability: Attributes of software that bear on the effort needed for modification, fault removal or for environmental change.

d) Testability: Attributes of software that bear on the effort needed for validating the modified software.

Portability: A set of attributes that bear on the ability of software to be transferred from one environment to another

a) Installability: Attributes of software that bear on the effort needed to install the software in specified environment.

b) Replaceability: Attributes of software that bear on the opportunity and effort of using it in the place of specified other software in the environment of that software,

c) Adaptability: Attributes of software that bear on the opportunity for its adaptation to different specified environments without applying other actions or means than those provide for this purpose.

d) Conformance: Attributes of software that bear on the opportunity and effort of using it in the place of specified other software in the environment of that software.