

---

# Quality Management of Software and Systems

## Terminology

### Contents

---

- ☐ System, technical system
- ☐ Quality, quality requirement, quality characteristic, quality measure
- ☐ Safety, technical safety
- ☐ Correctness, completeness
- ☐ Robustness
- ☐ Reliability, availability
- ☐ Failure, fault, error

## Definition of Terms

---

- ☐ System
  - Technical and organizational means for the autonomous fulfillment of a task (based on Birolini, ETH)
  - Generally, a system can consist of hardware, software, people (service and maintenance personnel) and logistic assistance
- ☐ Technical System
  - System where influences by people and logistics are ignored

## Definition of Terms

---

- ☐ Quality
  - Property of an entity concerning its qualification to fulfill defined and derived requirements (quality requirements) /DIN 55350-11/
- ☐ Quality Requirement
  - Total of single requirements of an entity which concern the property of this entity /DIN 55350-11/
- ☐ Quality Characteristic
  - Property of an entity on the basis of which its quality is described and estimated, but which makes no statement about the degree of fulfillment of the characteristic
  - A quality characteristic can be refined incrementally into partial characteristics
- ☐ Quality Measure
  - Measure which allows to draw conclusions on the fulfillment of specific quality characteristics

## Definition of Terms

---

### ☐ Safety

- State where the danger of a personal or property damage is reduced to an acceptable value (DIN EN ISO 8402)
- Birolini defines safety as a measure for the ability of an item to endanger neither persons, property nor the environment
- A distinction is drawn between the safety of a failure-free system (accident prevention) and the technical safety of the failure afflicted system

### ☐ Technical Safety

- Measure for the ability of a failure afflicted item to endanger neither persons, property nor the environment

## Definition of Terms

---

### ☐ Correctness

- Correctness has a binary character, i.e., an item is either correct or incorrect
- An fault-free realization is correct
- An artifact is correct if it is consistent to its specification
- If no specification exists for an artifact, correctness is not defined

### ☐ Completeness

- A system is functional complete, if all functions required in the specification are implemented. This concerns the treatment of normal cases as well as the interception of failure situations

## Definition of Terms

---

### ☐ Robustness

- Property to deliver an acceptable behavior also in exceptional situations (e.g. ability of a software to detect hardware failures)
- A correct system – as measured by the specification – can have a low robustness, actually
- Accordingly, robustness is rather a property of the specification than of the implementation
- A robust program is the result of the correct implementation of a good and complete specification
- Robustness has a gradual character

## Definition of Terms

---

### ☐ Reliability

- Part of the quality with regard to the behavior of an entity during or after given time periods with given working conditions (DIN 40041)
- Collective term for the description of the power concerning availability and its influencing factors: power concerning functionality, maintainability and maintainability support (DIN EN ISO 8402)
- Property of an entity regarding its qualification to fulfill the reliability requirements during or after given time periods with given application requirements (DIN ISO 9000)
- 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 (based on Birolini, ETH)

### ☐ Availability

- Measure for the ability of an item to be functional at a given time

## Definition of Terms

---

### ☐ Failure, Fault, Error

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