Chapter 1

Introduction
1.1 About this Course

1.2 Quality Principles

1.3 Quality management
Software quality matters

WE STILL HAVE TOO MANY SOFTWARE FAULTS. WE’LL MISS OUR SHIP DATE.

MOVE THE LIST OF FAULTS TO THE “FUTURE DEVELOPMENT” COLUMN AND SHIP IT.

90% OF THIS JOB IS FIGURING OUT WHAT TO CALL STUFF.
Contents

- This is an advanced course about software quality
- It covers selected aspects of software quality, in particular
  - Model Checking
  - Advanced testing
  - Debugging
  - Process quality
  - External and internal product quality
  - Dependability
  - Quality in agile development
- Familiarity with the basics of software quality is a prerequisite (see next slide)
Prerequisites

❍ **Familiarity with the basics** of software quality, as covered in a course or textbook in Software Engineering is expected.

❍ **From the Software Engineering course at UZH, we expect** that you are familiar with these chapters:
  - Kapitel 2: Ziele und Qualität
  - Kapitel 7: Validierung und Verifikation
  - Kapitel 8: Testen von Software
  - Kapitel 9: Reviews
  - Kapitel 10: Messen von Software
  - Kapitel 11: Statische Analyse
  - Kapitel 16: Software-Qualitätsmanagement
  - Kapitel 17: Bewertung und Verbesserung von Prozessen und Qualität
Learning Goals

❖ Students acquire knowledge and skills in advanced topics of software quality, thus enabling them to
  ● analyze, assess and improve software quality
  ● develop high-quality software

❖ Students deepen and extend their knowledge in
  ● Advanced testing techniques
  ● Quality management
  ● Product and process quality

❖ Students learn about
  ● Model Checking as an important verification procedure
  ● Systematic debugging as a means for locating causes of errors
Tentative Schedule

2017-02-20  Introduction
2017-02-27  Model Checking
2017-03-06  Advanced Testing Techniques
2017-03-13  Debugging; Discussion of assignment 1
2017-03-20  Process Quality
2017-03-27  Product Quality; Quality in Agile Development,
2017-04-03  Discussion of assignment 2
2017-05-08  Final exam
Homework assignment (exercise) schedule

2017-02-27  Introduction to assignment 1
2017-03-06  Deadline for submission
2017-03-13  Discussion of assignment 1, Introduction to assignment 2

2017-03-20  Deadline for submission
2017-04-03  Discussion of assignment 2
Passing the course

Pass the two assignments (pass/fail)
and
Pass the final exam (graded)

The course grade will be the grade achieved in the final exam
1.1 About this Course

1.2 Quality Principles

1.3 Quality management
Intuitive notion of quality

What’s your personal intuitive notion of quality?

Intuitively, quality is typically associated with

- **High-grade** products or services
- **Durable** products, *carefully made* according to highest standards
Industrial notion of quality

Quality – The degree to which a set of inherent characteristics of an entity fulfills requirements. [ISO 9000:2005]

Inherent characteristic – A characteristic that forms a constituent part of an entity, as opposed to assigned characteristics

Entity – A product, service, process, system, organization, ...

Example: A drug
- Inherent characteristic: Ingredients
- Explicitly assigned: Price
Remarks

- The industrial notion of quality does not fully match the intuitive one.

- Quality means satisfying requirements. Requirements may be explicitly stated or implicitly given by a shared notion.

- Quality is no absolute degree of goodness.

- Considering quality merely as fitness for purpose or customer satisfaction falls short of the full meaning.

- Quality does not emerge by itself. Quality must be defined and explicitly created.
Software quality

- The entity (that the inherent characteristics of which shall fulfill requirements) is a **software system** or **component**
- Software is **different:**
  - not tangible
  - manifests only in **effects** and **documentation**, especially the **source code**
External vs. internal software quality

The pragmatic view

- **External** quality is quality as perceived by stakeholders
- **Internal** quality is the quality of the software, particularly of the source code that eventually delivers external quality
External vs. internal software quality – 2

The ISO standards’ view

The ISO software quality standards distinguish quality from a measurement viewpoint (cf. Chapter 6)

- Internal measures: internal quality
- External measures: external quality
- Usage measures: quality in use

[ISO/IEC 25010:2011]
1.1 About this Course

1.2 Quality Principles

1.3 Quality management
Principles of modern quality management

- Involvement of people: everybody takes full responsibility
- Customer focus
- Process-oriented, systemic approach

ISO 9000:2005
Quality management terminology

Quality management – Coordinated activities to direct and control an organization with regard to quality

Quality planning – Part of quality management focused on setting quality objectives and specifying necessary operational processes and related resources to fulfill the quality objectives

Quality control – Part of quality management focused on fulfilling quality requirements

[ISO 9000:2005]
Quality management terminology – 2

Quality assurance – Part of quality management focused on providing confidence that quality requirements will be fulfilled

Quality improvement – Part of quality management focused on increasing the ability to fulfill quality requirements
Procedures of quality management

**Quality planning**
Definition of requirements: We want to achieve this!

**Quality control**
- constructive: this is how we need to work.
- analytic: did we work correctly? (quality checking)

**Quality improvement**
- What works how well?
- Where can we spot weaknesses?
- We want to improve this!
Constructive quality control

- Using processes that prevent/avoid faults
- Rigorous and early quality checking
- Continuous improvement of quality and processes
Analytic quality control

- Checking the **product**
  - Static methods
    - Reviewing
    - Static analysis
    - Formal verification
    - Model checking
  - Dynamic methods
    - Testing
    - Simulation
    - Prototyping

- Checking the **process(es)**
  - Audits
  - Process capability assessment
Analytic techniques: a quiz

○ Testing
  ● Why do we test?
  ● What are the ingredients of a systematic test?
  ● Which forms of testing do you know?

○ Reviewing
  ● What is a review?
  ● Which forms of reviews do you know?
  ● What are the rules for systematic reviewing?

○ Static analysis
  ● What is static analysis?
  ● What are typical things one can check with static analysis?
Analytic techniques: a quiz – 2

❍ Prototyping
  ● What is a software prototype?
  ● Which forms of prototyping do you know and what do they serve for?
  ● Throw-away prototypes vs. evolutionary prototyping?

❍ Auditing
  ● What is an audit?
  ● What does auditing mean in the context of software quality?
  ● Which forms of software quality audits do you know?
  ● How is an audit carried out?
Quality improvement

Just fixing quality defects is not enough

- **Necessary** for achieving product quality in software
- However: frequently fixes *symptoms* only

→ **Systemic approach needed**
Quality improvement – 2

A systemic approach to quality improvement

Modify

• quality related processes and/or

• the quality management system

Based on

○ Systematic evaluation of

  ● Observed errors/failures and their causes
  ● Findings in reviews

○ Product and process measurement

○ Findings in audits

→ Process improvement
Process improvement

The Deming cycle: Plan-Do-Check-Act (Deming 1986)

- Plan
  - Design
  - Document

- Do
  - Train
  - Institutionalize
  - Perform

- Check
  - Measure
  - Analyze
  - Control

- Act
  - Adapt
  - Eliminate weaknesses
  - Automate

- Measure
  - Control
References


