



University of  
Zurich<sup>UZH</sup>

Department of Informatics

Martin Glinz

# Software Quality

Chapter 1

## Introduction

# 1.1 About this Course

---

1.2 Quality Principles

1.3 Quality management



# Software quality matters

---



# 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

---

- 2016-02-22 Introduction
- 2016-02-29 Model Checking
- 2016-03-07 Advanced Testing Techniques
- 2016-03-14 No lecture; Discussion of assignment 1
- 2016-03-21 Debugging
- 2016-03-28 No class (Easter break)
- 2016-04-04 Process Quality, Discussion of assignment 2
- 2016-04-11 Product Quality; Quality in Agile Development
  
- 2016-05-02 Final exam

# Homework assignment (exercise) schedule

---

2016-02-29 Introduction to assignment 1

2016-03-09 Deadline for submission

2016-03-14 Discussion of assignment 1, Introduction to assignment 2

2016-03-30 Deadline for submission

2016-04-04 Discussion of assignment 2



# Passing the course

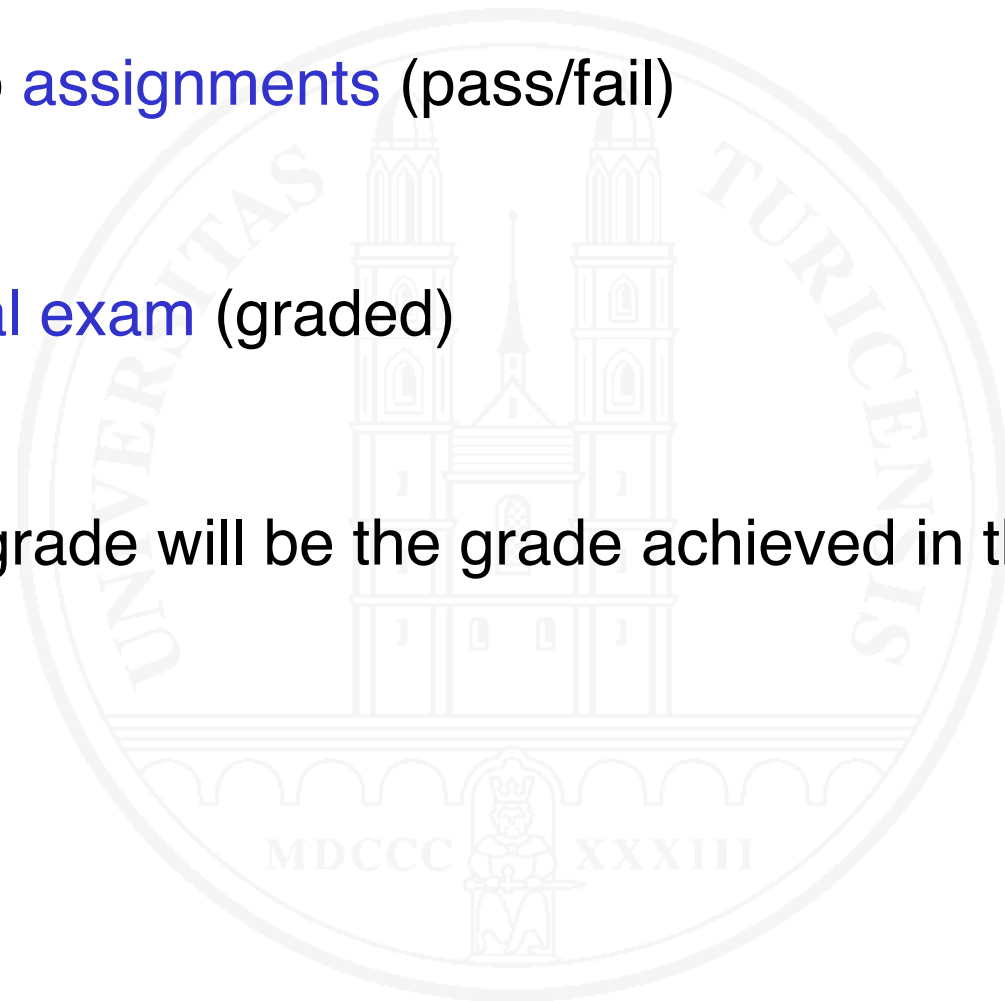
---

Pass the two **assignments** (pass/fail)

and

Pass the **final exam** (graded)

The course grade will be the grade achieved in the final exam

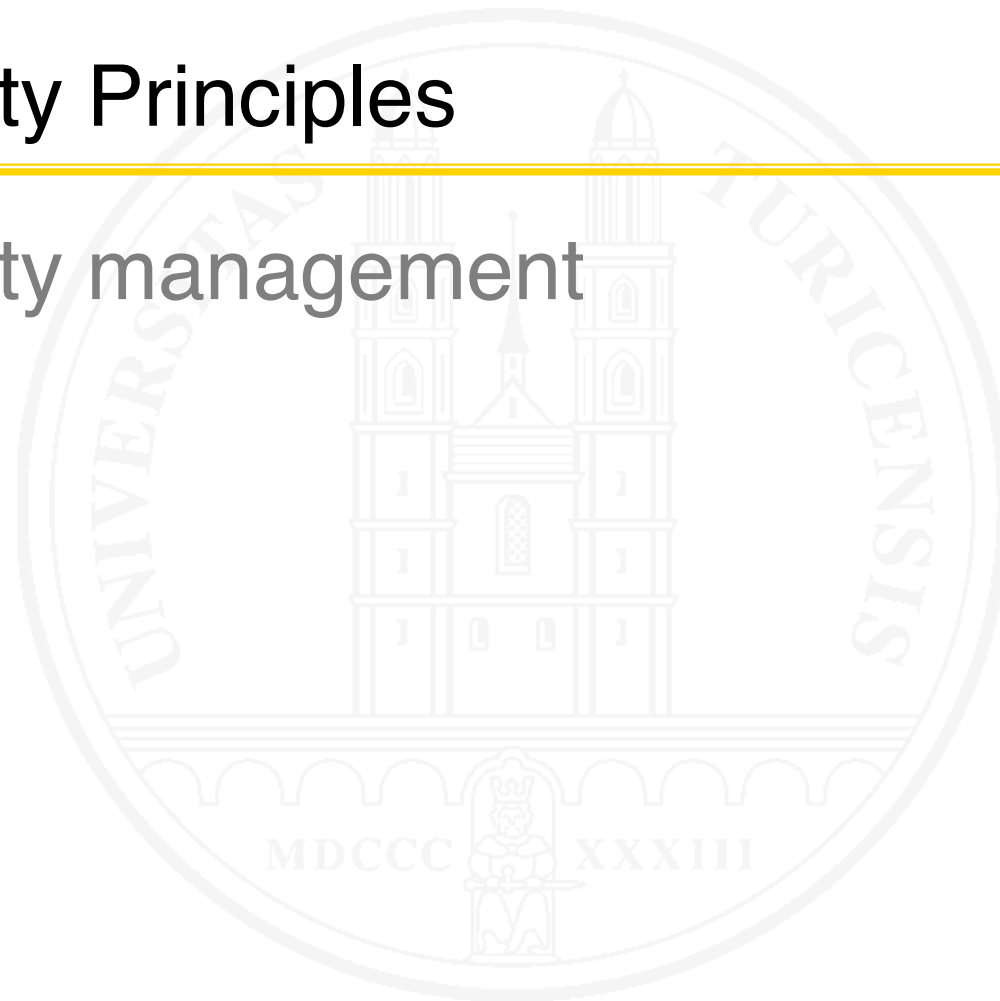


## 1.1 About this Course

## 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

---

[ISO/IEC 25010:2011]

## 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**

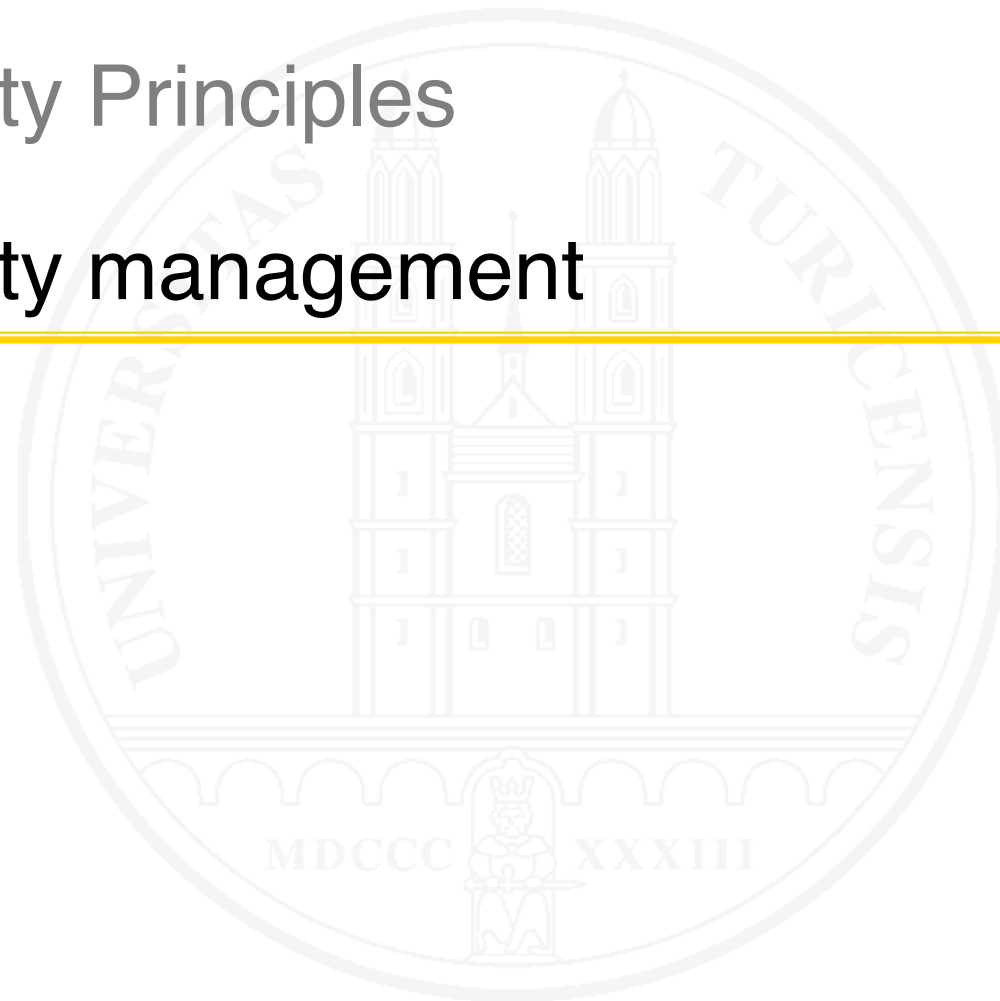


1.1 About this Course

1.2 Quality Principles

1.3 Quality management

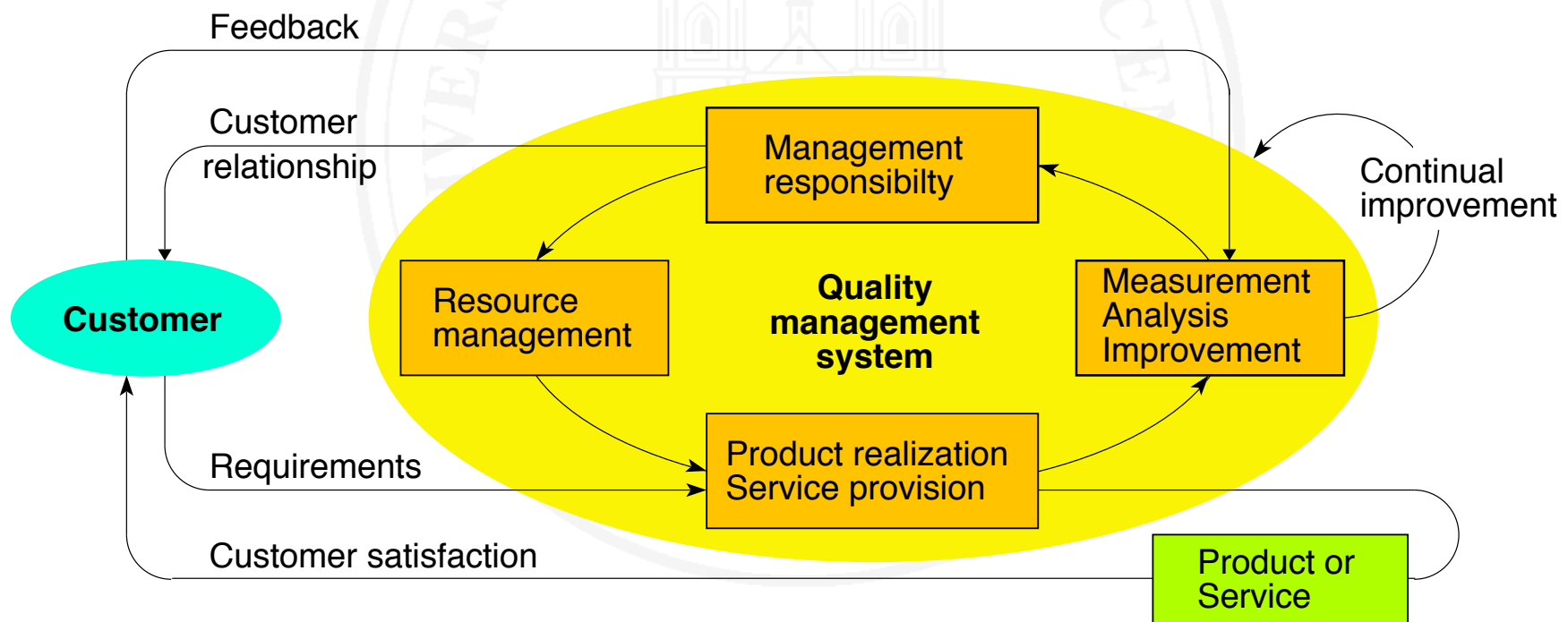
---



# Principles of modern quality management

[ISO 9000:2005]

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



# Quality management terminology

---

[ISO 9000:2005]

**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**

# Quality management terminology – 2

---

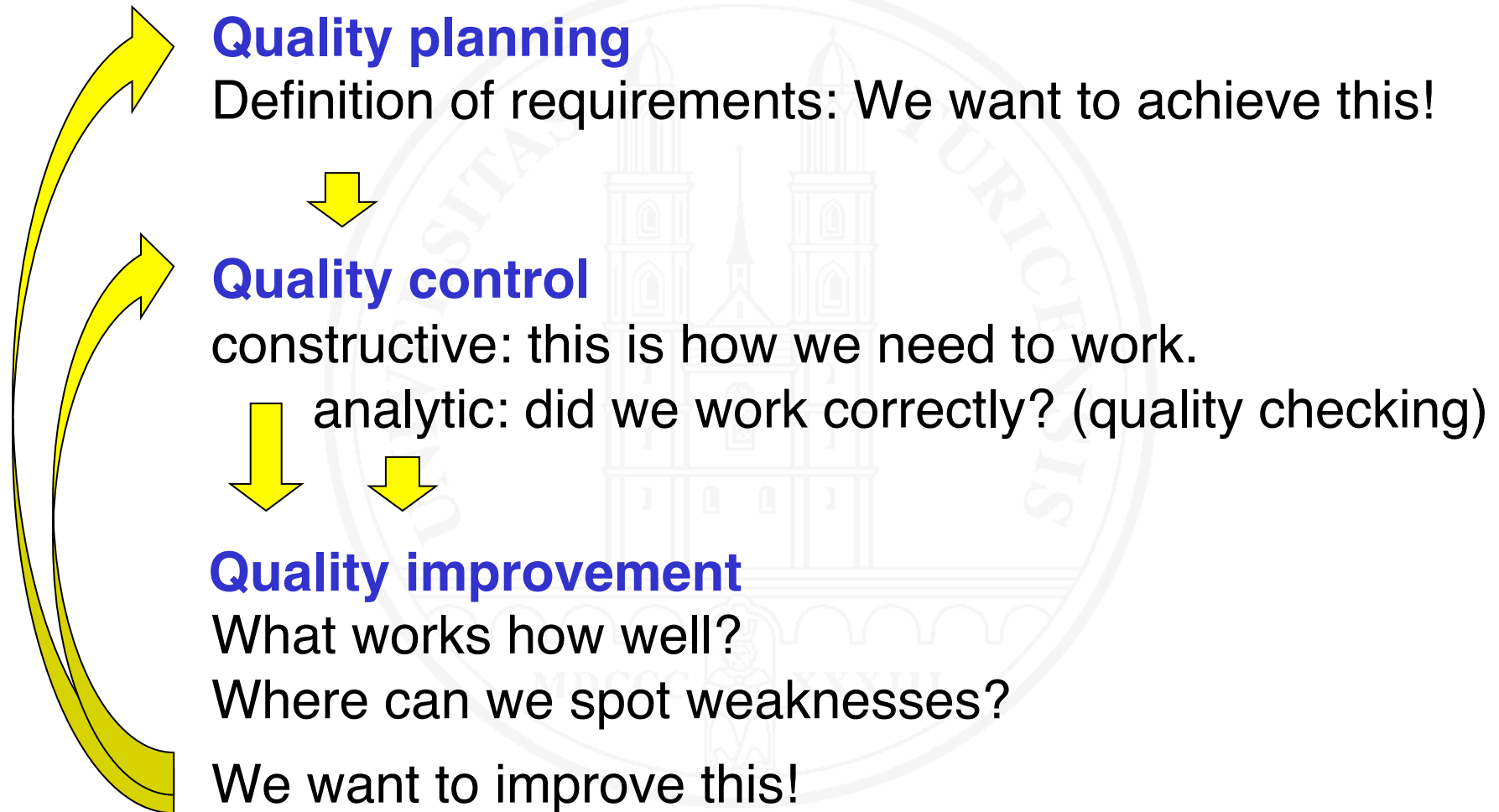
[ISO 9000:2005]

**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

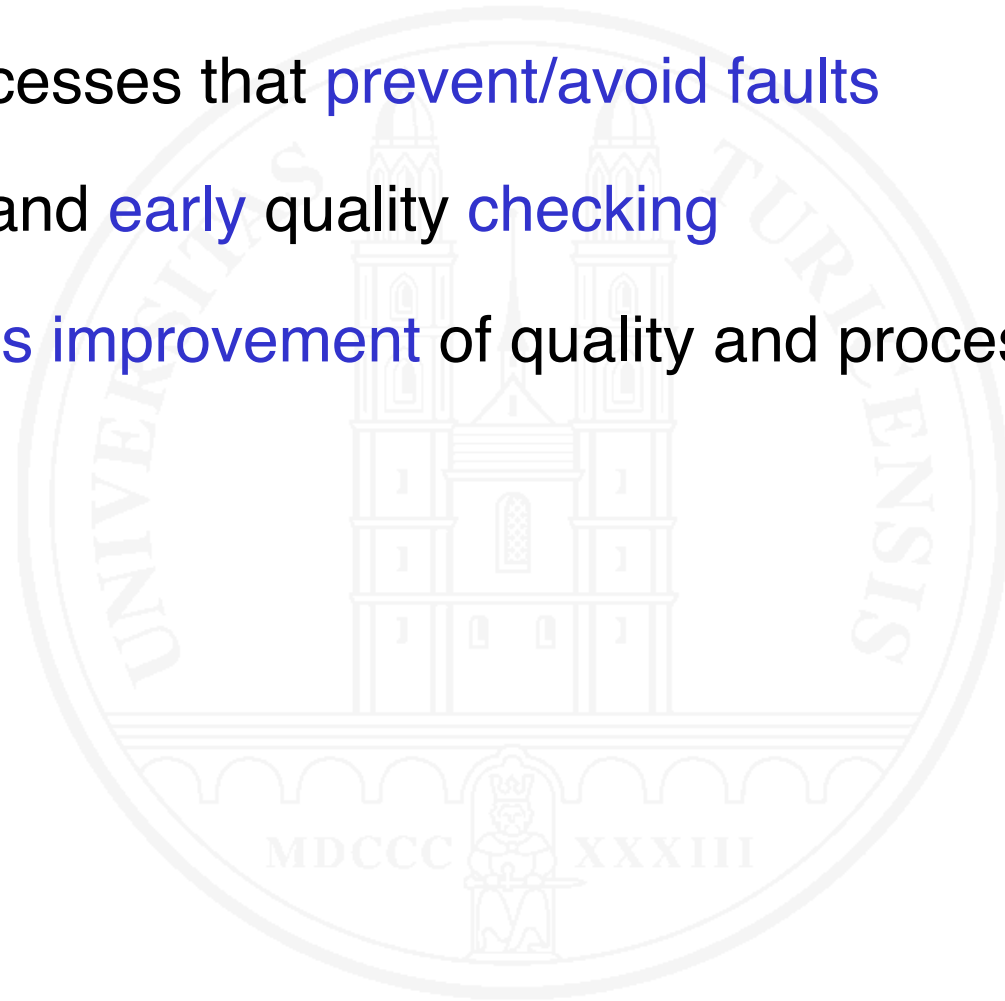
---



# 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**

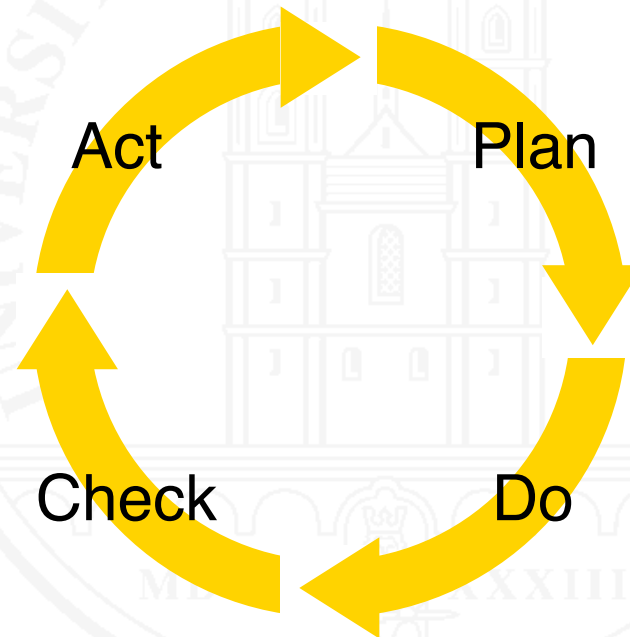
# Process improvement

---

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

- Adapt
- Eliminate weaknesses
- Automate

- Measure
- Analyze
- Control



- Design
- Document
- Train
- Institutionalize
- Perform

# References

---

W.E. Deming (1986). *Out of the Crisis*. Cambridge, Mass.: M.I.T. Press.

M. Glinz, T. Fritz (2015). *Software Engineering*. Course Slides. University of Zurich.

ISO (2005). *Quality management systems – Fundamentals and vocabulary*. International standard ISO 9000:2005.

ISO (2008). *Quality management systems – Requirements*. International standard ISO 9001:2008.

ISO (2011). *Managing for the sustained success of an organization – A quality management approach*. International Standard ISO 9004:2011.

ISO/IEC (2011). *Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – System and software quality models*. International Standard ISO/IEC 25010:2011

B. Kitchenham, S.L. Pfleeger (1996). Software Quality: The Elusive Target. *IEEE Software* **13**(1):12–21.

P. Liggesmeyer (2002). *Software-Qualität: Testen, Analysieren und Verifizieren von Software*. Heidelberg; Berlin: Spektrum Akademischer Verlag.

K. Schneider (2012). *Abenteuer Softwarequalität. 2. Auflage*. Heidelberg: dpunkt.

M. Wieczorek, D. Meyerhoff (eds.) (2001). *Software Quality: State of the Art in Management, Testing, and Tools*. Berlin: Springer.