Requirements Engineering I

Martin Glinz

Department of Informatics, University of Zurich
www.ifi.uzh.ch/~glinz

University of Zurich UZH

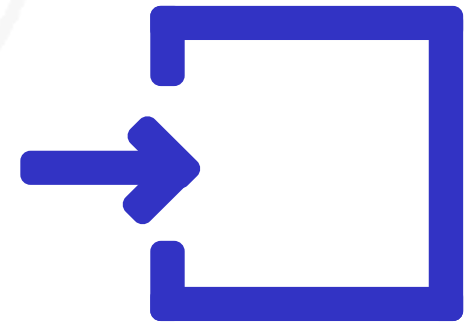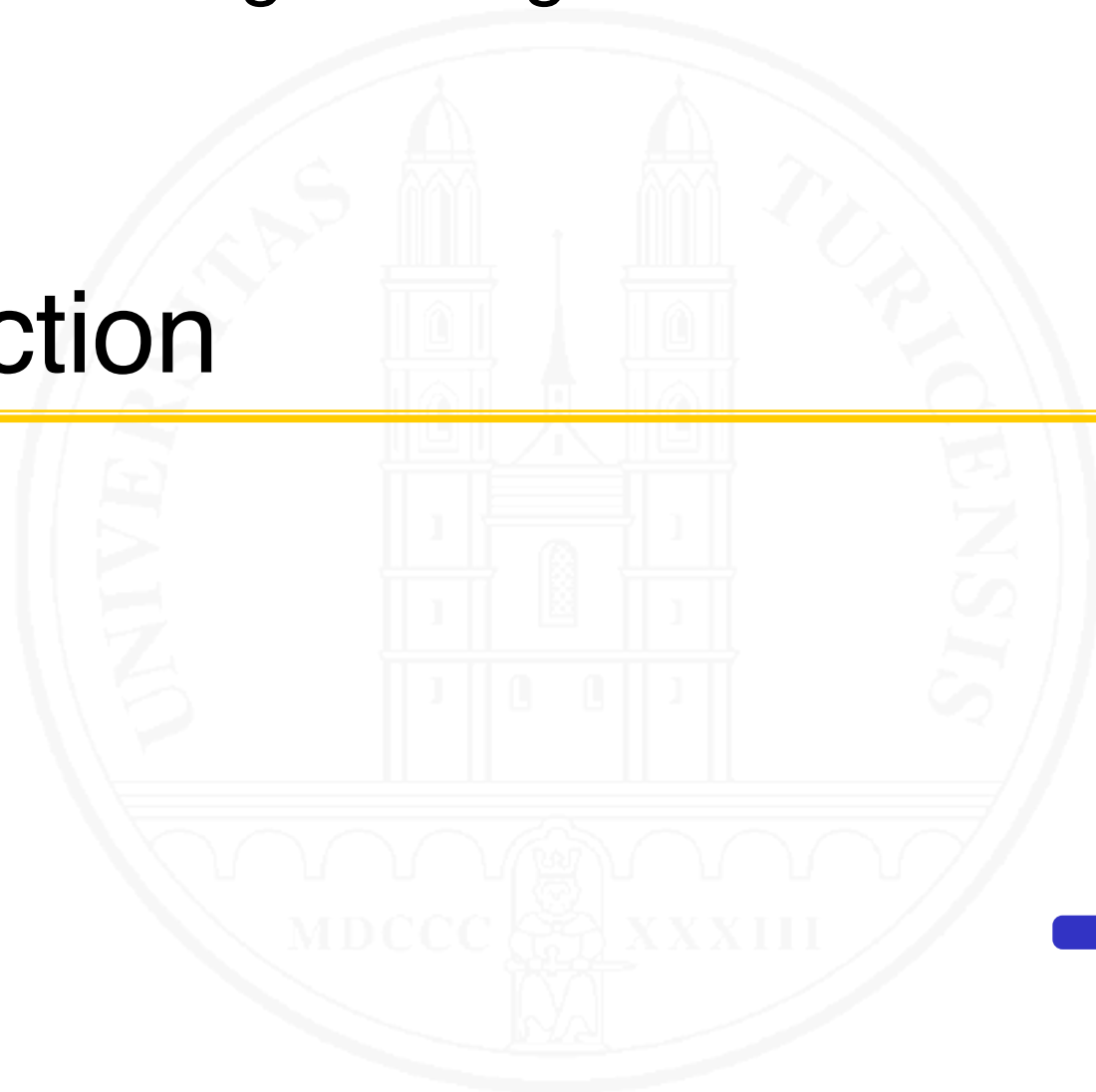**Department of Informatics**

Requirements
Engineering
Research
Group

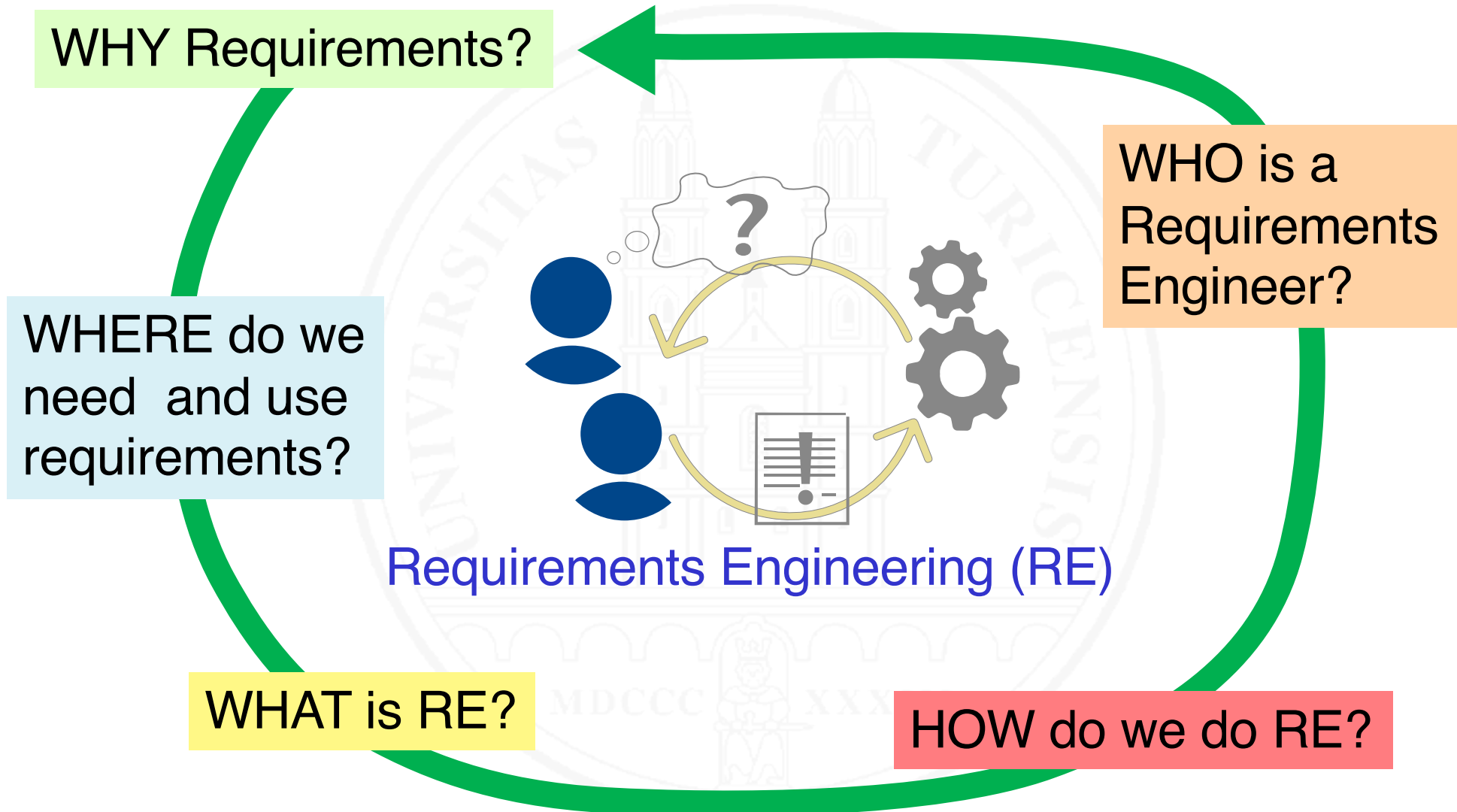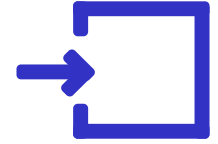**RE**
RG

Requirements Engineering I

Chapter 1

Introduction

Chapter roadmap

WHY Requirements?

WHO is a Requirements Engineer?

WHERE do we need and use requirements?

Requirements Engineering (RE)

WHAT is RE?

HOW do we do RE?

A case: Sales and customer support platform

Context: Corporate customer business in a a big international insurance company

Current state:

- Insurance agents have a laptop with
  - Powerpoint presentations of products
  - PDFs with terms and conditions
  - Editor for filling-in applications
  - Front-end app for querying the host system
  - Access to CRM system
- No web interface for corporate customers

Problems: Complicated, missing features, inflexible

Decision: Build a new platform for sales and customer support

Challenge: What, precisely, shall be built?

A communication and understanding problem

WHY

Need, Problem

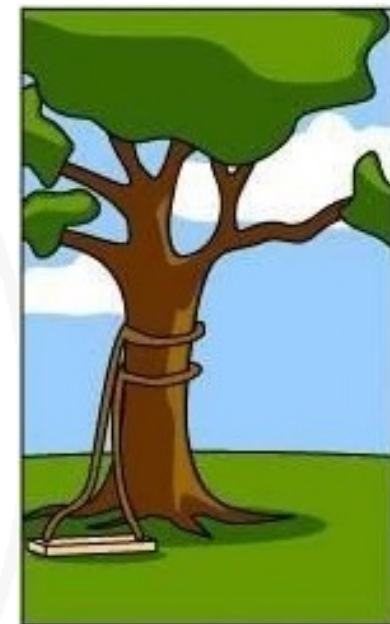What the customer wanted

Analysis

What the analyst understood

Design

What the architect designed

Deployed System

What the programmers implemented

We need to know the requirements.

DEFINITION. Requirement –

1. A need perceived by a stakeholder.

2. A capability or property that a system shall have.

3. A documented representation of a need, capability or property.

[Glinz 2020]

Note: A Stakeholder is a person or organization who influences a system's requirements or who is impacted by that system; see Chapter 2.

Where do we need and use requirements?

In principle: For any kind of system

In particular: for software-intensive systems

More than just software:

Cyber-physical systems contain both software and physical components

Socio-technical systems span software, hardware, people and organizational aspects

Terminology: What is a system?

DEFINITION. System – 1. A principle for ordering and structuring. 2. A coherent, delimitable set of elements that – by coordinated action – achieve some purpose.

❍ A system may comprise other systems

❍ The purpose achieved by a system may be delivered by
   ● deploying it at the place(s) where it is used
   ● selling/providing it as a product to its users
   ● having providers who offer the system's capabilities as services to users

❍ Requirements Engineering is primarily concerned with systems in which software plays a major role

Forms of occurrence

Requirements occur in various forms:

System requirements – How a system shall work and behave

Stakeholder requirements – Stakeholders' desires and needs from a stakeholder perspective

   User requirements – A subset of the stakeholder requirements

Domain requirements – Required domain properties of a socio-technical or cyber-physical system

Business requirements – Focus on business goals, objectives and needs of an organization

→ Chapter 3

A sample problem

A ski resort operates several chairlifts. They need to renovate the access control to the chairlifts.

The idea: Skiers buy RFID-equipped day access cards. Access to the lifts is controlled by RFID-enabled turnstiles. Whenever a turnstile senses a valid access card, it unlocks the turnstile for one turn, so that the skier can pass.
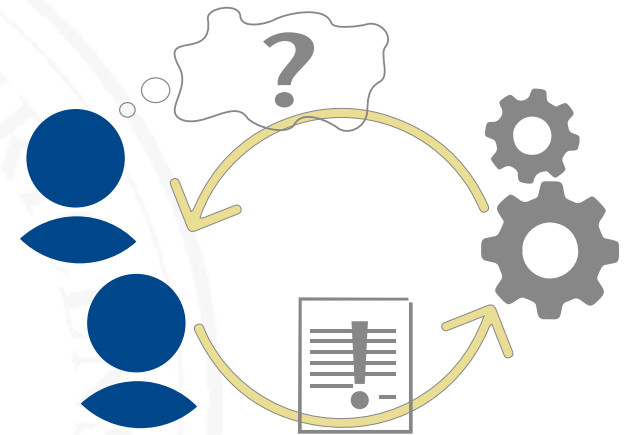
The task: Build a software-controlled system for managing the access of skiers to the chairlifts.

When building such a system...

...   we need to understand the problem and agree upon what to build.

→   Requirements Engineering

"The hardest single part of building a software system is deciding precisely what to build."
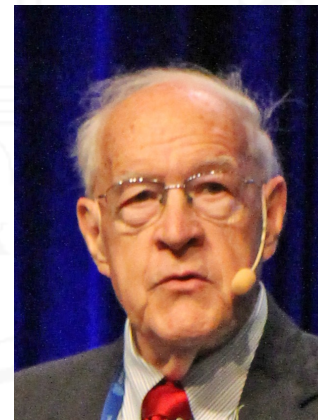*Frederick P. Brooks (1987)*

Photo: Martin Glinz (2018)

What is Requirements Engineering (RE)?

Since its inception in the 1970ies, the notion of Requirements Engineering has evolved:

From a purely technical definition

to the contemporary one which focuses on satisfying stakeholders' needs and reducing development risk.

We illustrate this evolution with a series of definitions.

The traditional definition of RE

DEFINITION. Requirements Engineering (RE) [Traditional] – The application of a systematic, disciplined, quantifiable approach to the specification and management of requirements; that is the application of engineering to requirements.

[Adapted from the definition of Software Engineering in IEEE 610.12-1990]

Metaphor: upfront engineering

Goal: complete, unambiguous requirements prior to design

Smells: paper, process

Reality check: Does this always work?

Wait a minute – it's about customers' needs

DEFINITION. Requirements Engineering [Customer-oriented] – Understanding and documenting the customers' desires and needs.

[Glinz 1998, inspired by Gause and Weinberg (1989)]

Metaphor: Customer satisfaction

Goal: Understand the customer

Reality check:

(1)   Why not just code what the customer desires and needs?

(2)   Who is "the customer"?

Where's the value?

DEFINITION. Requirements Engineering [Risk-oriented] – Specifying and managing requirements to minimize the risk of delivering a system that does not meet the stakeholders' desires and needs.

[Glinz 2003]

Metaphor: Balancing effort and value

Goal: Mitigate risk

Risk-based RE

"We have no time for a complete specification."

"This is too expensive!"

"We're agile, so rough stories suffice."

⬇ Wrong approach

Right question: "How much RE do we need such that the risk of deploying the wrong system becomes acceptable?"

Rule:

The *effort* spent for Requirements Engineering shall be *inversely proportional* to the *risk* that one is willing to take.

The contemporary definition of RE

DEFINITION. Requirements Engineering – The systematic and disciplined approach to the specification and management of requirements with the goal of understanding the stakeholders' desires and needs and minimizing the risk of delivering a system that does not meet these desires and needs.

[Glinz (2020); for the definition of 'stakeholder' see Chapter 2]

A note on terminology

- ❍ Lots of sources for today's terminology
  - Textbooks and articles about RE
  - IEEE 610.12 (1990) – a slightly aged glossary of software engineering terminology
  - IEEE 830-1998 – an outdated, but still cited RE standard
  - ISO/IEC/IEEE 29148 (2018) – a new, but still rather unknown RE standard; provides definitions of selected terms, some of them being rather uncommon
  - ISO/IEC/IEEE ISO/IEC/IEEE 24765 (2017) – compiles definitions from various other standards
  - IREB Glossary [Glinz 2020] – influential through IREB's certification activities; used as a terminology basis in this course

Mini-Exercise

Recall (without looking it up) the three main constituents of contemporary Requirements Engineering.

The systematic and disciplined approach to the specification and management of requirements with the goal of

understanding the stakeholders' desires and needs and

minimizing the risk of delivering a system that does not meet these desires and needs.

How do we do RE?

Four major tasks:

❍ Eliciting the requirements

❍ Analyzing and documenting the requirements

❍ Validating the requirements

❍ Managing and evolving the requirements

The requirements engineer   WHO

❍ Mostly a role, not a job title

❍ The role is part of many job functions: business analyst, system analyst, application engineer, software engineer, systems engineer, product owner, ...

❍ People act as requirements engineers if they

- elicit, document, validate and manage requirements,

- have in-depth knowledge of Requirements Engineering, enabling them to define RE processes, select appropriate RE practices and apply them properly,

- are able to bridge the gap between the problem and potential solutions

Is Requirements Engineering worthwhile?
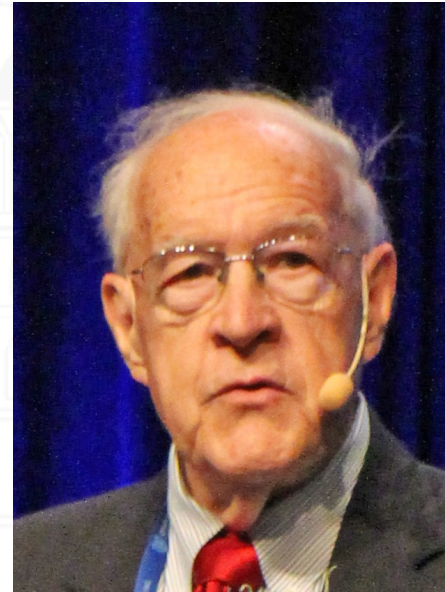
WHY

Photo: Hamilton Richards (2002)

Photo: Martin Glinz (2018)

"Not worthwhile"
Edsger W. Dijkstra (1993*)

"The hardest single part of building a software system"
Frederick P. Brooks (1987)

*Personal communication

Is Requirements Engineering worthwhile?

Systems development is an expensive and risky business.

Requirements Engineering

❍ helps understand the problem

❍ reduces the risk of failure or costly modifications in later development stages

❍ provides a proper basis for estimating development effort and cost

❍ is a prerequisite for testing the developed system properly

A note about cost and benefit

Requirements Engineering contributes to

❍ Reducing error and rework cost

❍ Managing the development risk
  - Meet stakeholders' desires and needs
  - Reliable estimates for deadlines and cost

☞ The economic effects of Requirements Engineering are (almost ever) indirect ones; RE as such just costs!

Lower cost

Higher benefit