Course introduction

Kristian Sandahl/Daniel Ståhl/Dániel Varró

LINKÖPING UNIVERSITY

Agenda:

Presentation

Problems with software

What is Software Engineering?

Course Goals

Labs and Projects (TDDC88/725G64)

Studying Software Engineering

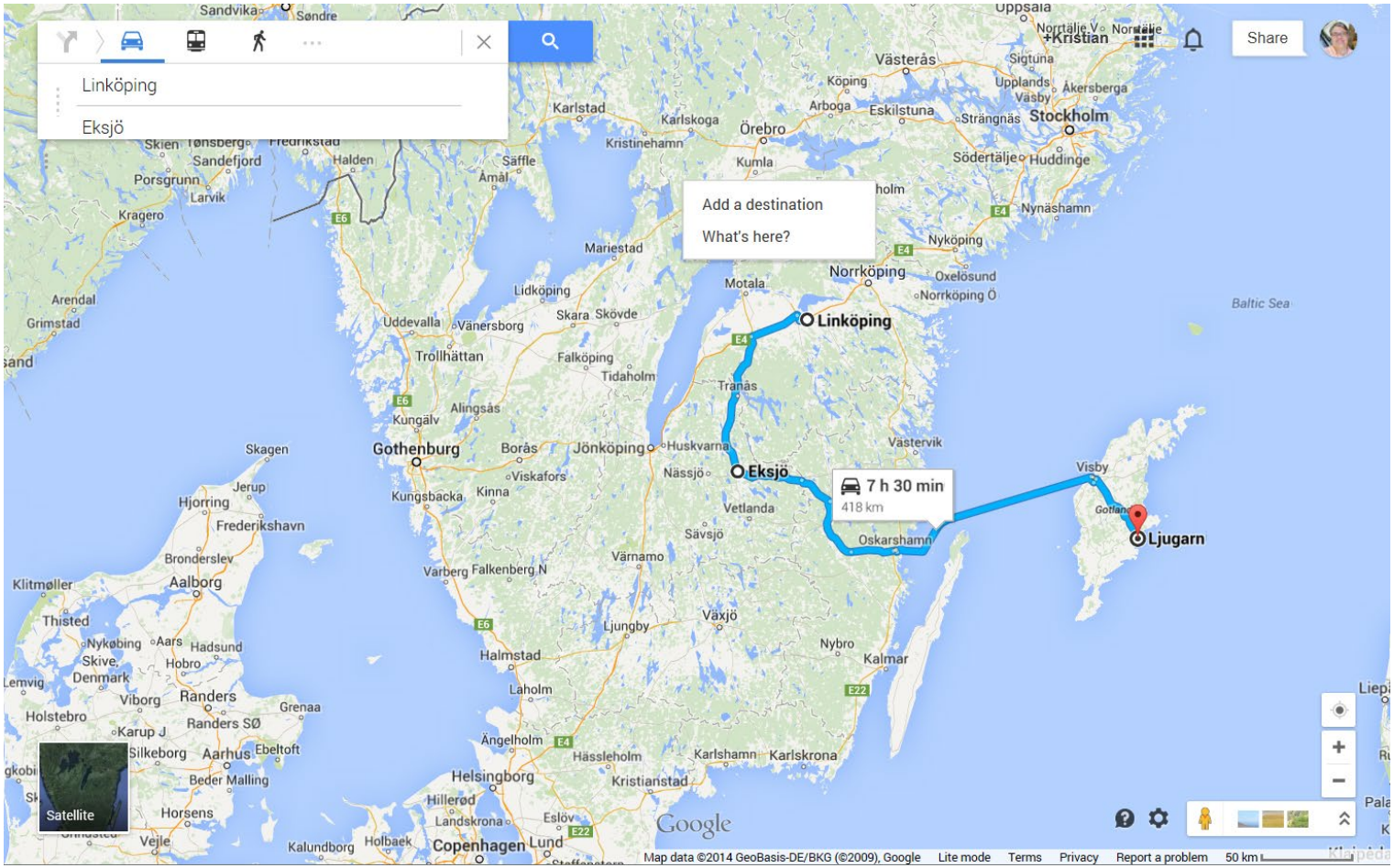
Contents of the theory part

Communication

Examination

Changes since last year

Where does Kristian come from?



Who is Kristian?

- Professor of Software Engineering since 2001
- I love studying (large) software development organizations and evaluate practices of Software Engineering
- I love teaching project courses, but I'm not a faithful constructivist
- My mission: To bring some critical thinking and common sense to the world
- Background: Civ ing (D) -83; PhD -92
  Epitec: spin-off company 86-87
  Ericsson research: 95-01

Who is Daniel Ståhl?

DANIEL STÅHL

MSc 2007
Linköping University

Joined Ericsson in 2009

PhD, 2017
University of Groningen

Joined LiU in 2019

ERICSSON

Developer
Architect
Continuous Practices

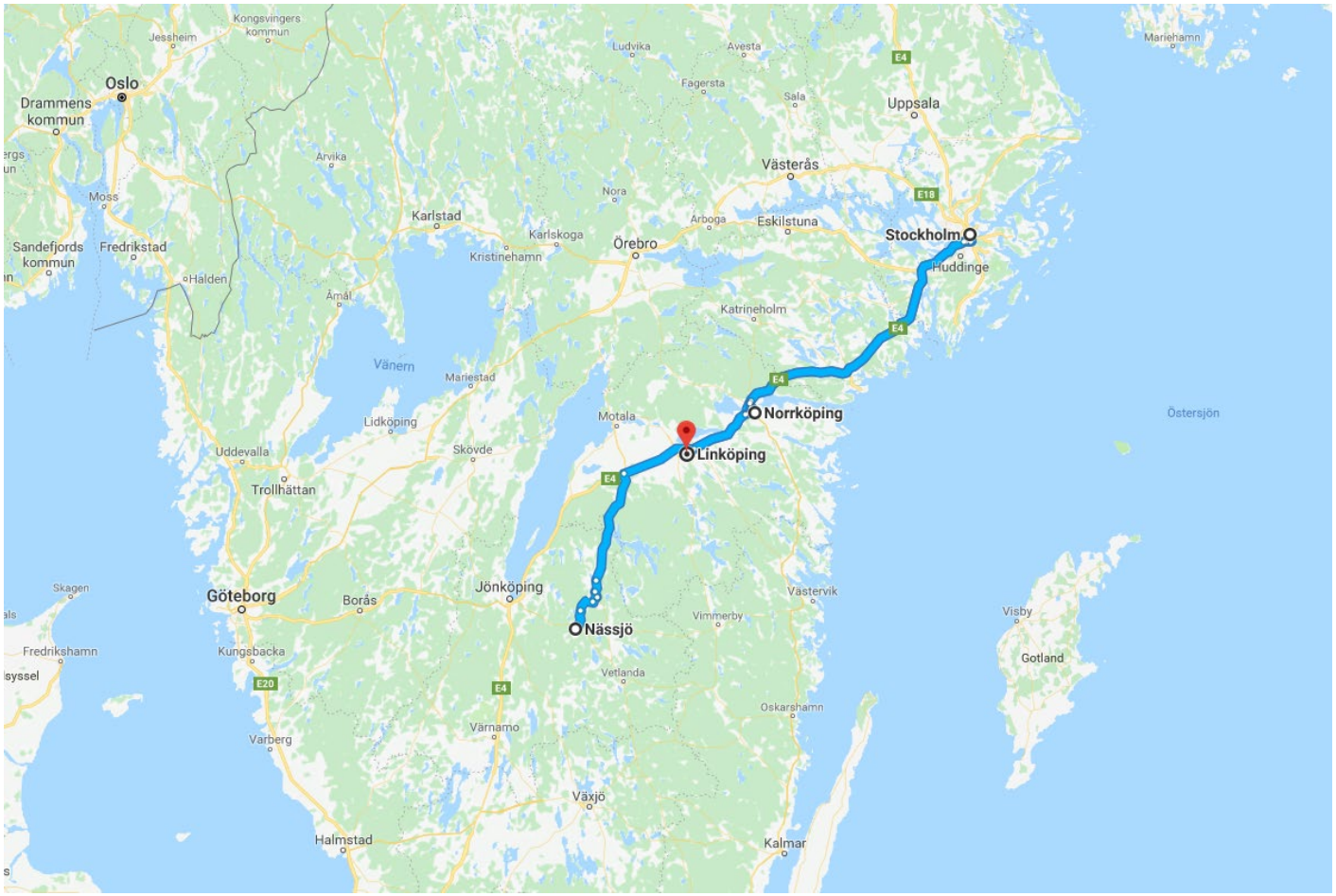
…

AI Strategy
AI Governance
Research supervision

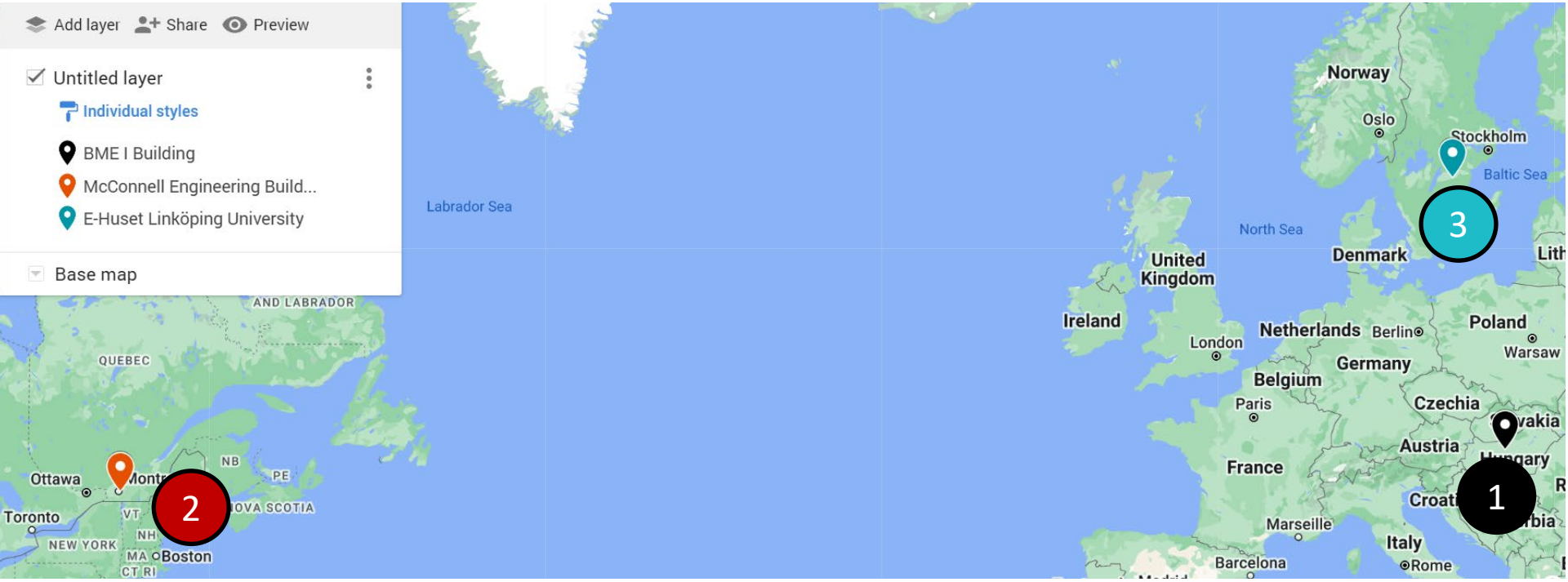
RESEARCH, WRITING

Continuous integration
Continuous delivery
Continuous deployment
Large-scale testing
Development practices
Teaming
Mob programming

Where does Daniel come from?

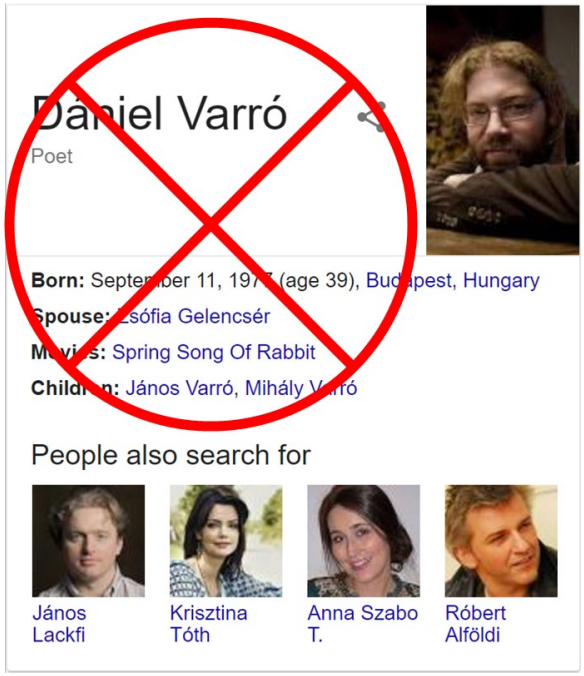


Where does DánielV come from?



Who is DanielV?

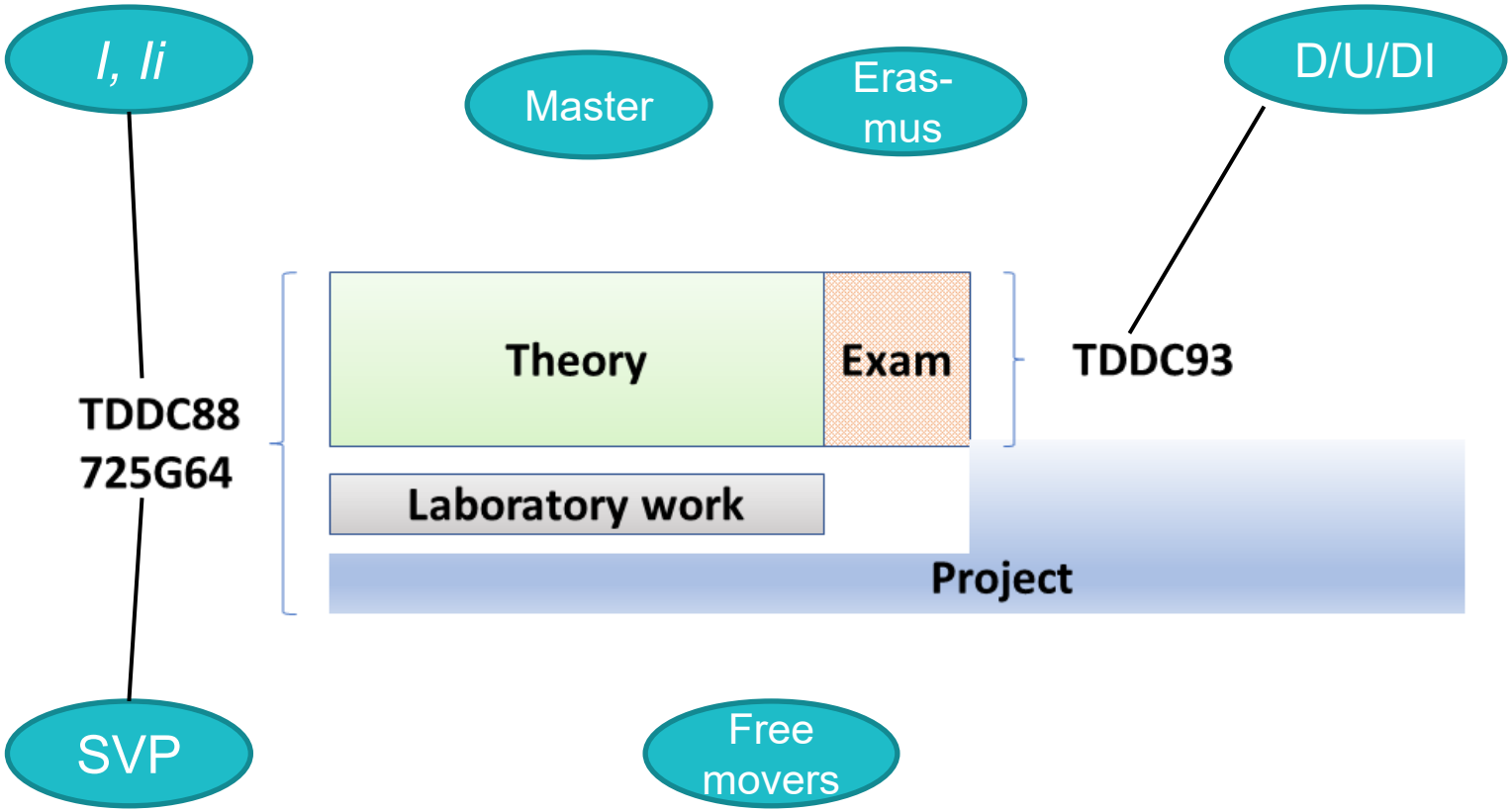
Who am I **NOT**?



Better findings:

- Professor of Software Engineering
  - BME (Budapest, Hungary)
  - McGill (Montreal, Canada)
  - LiU (Linköping, Sweden)
- Research positions
  - Lendület Research Chair (MTA Hungary)
  - WASP professor
- (Co-)Supervisor of 18 defended PhDs
- Steering committee vice-chair of MODELS
- Co-Founder of two spin-offs
  - IncQuery Labs, OptXware

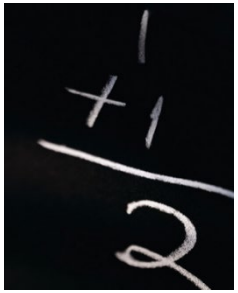
Who are you?


I, Ii
Master
Eras-
mus
D/U/DI
Theory
Exam
TDDC93
TDDC88
725G64
Laboratory work
Project
SVP
Free
movers

Why should I take this course?

Well, I have to, it's compulsory...

This is a rather pragmatic course, concerning **real problem** in the **industry**



It is **not** a hard mathematical course but requires some discipline and analysis

Are you going to work in the software industry?

Agenda:

~~Presentation~~

Problems with software

What is Software Engineering?

Course Goals

Labs and Projects (TDDC88/725G64)

Studying Software Engineering

Contents of the theory part

Communication

Examination

Changes since last year

If yes, how would you avoid disasters?

- Geppert, L (2004) Lost Radio Contact Leaves Pilots On Their Own, *IEEE Spectrum*, Nov 2004
- A commercial blog: https://www.computerworld.com/article/3412197/top-software-failures-in-recent-history.html

2^{32} milli-seconds is a looong time?

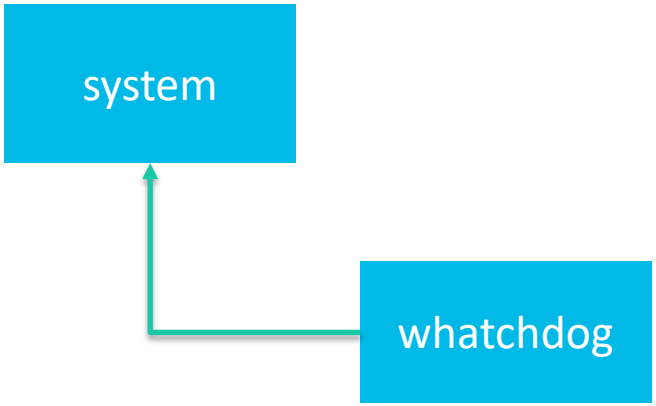
2^{32} ms ≈

$4.3 \cdot 10^6$ s ≈

$7.2 \cdot 10^4$ min ≈

1 193 h ≈

49.7 days



50 days ago was

2024-07-15

Every problem is a people problem...

UNDERSTANDING THE CUSTOMER

Much of the time, your customers won't know what they want themselves.

ANTICIPATING THE USER

Who is the end user? What would be valuable to them? Do they even know that themselves?

COLLABORATING WITH YOUR COLLEAGUES

How do you work alongside with your colleagues in the same project without stepping on one another's toes?

PLANNING AND LEADING

How do you lead, how do you delegate, and how do you communicate a common vision?

... and also a technology problem

UNDERSTANDING THE CUSTOMER

What are the technology choices that will enable the customer's long-term ambitions?

ANTICIPATING THE USER

What are the design patterns that will create a superior user experience? What are the tools and frameworks that will enable that?

COLLABORATING WITH YOUR COLLEAGUES

How do you set up an effective and efficient development environment? What does your tool chain look like? How do you enable transparency, trouble-shooting and build confidence?

PLANNING AND LEADING

How do you break down high-level ambitions into manageable tasks? How can you collect the necessary data to make confident predictions?

Agenda:

~~Presentation~~

~~Problems with software~~

What is Software Engineering?

Course Goals

Labs and Projects (TDDC88/725G64)

Studying Software Engineering

Contents of the theory part

Communication

Examination

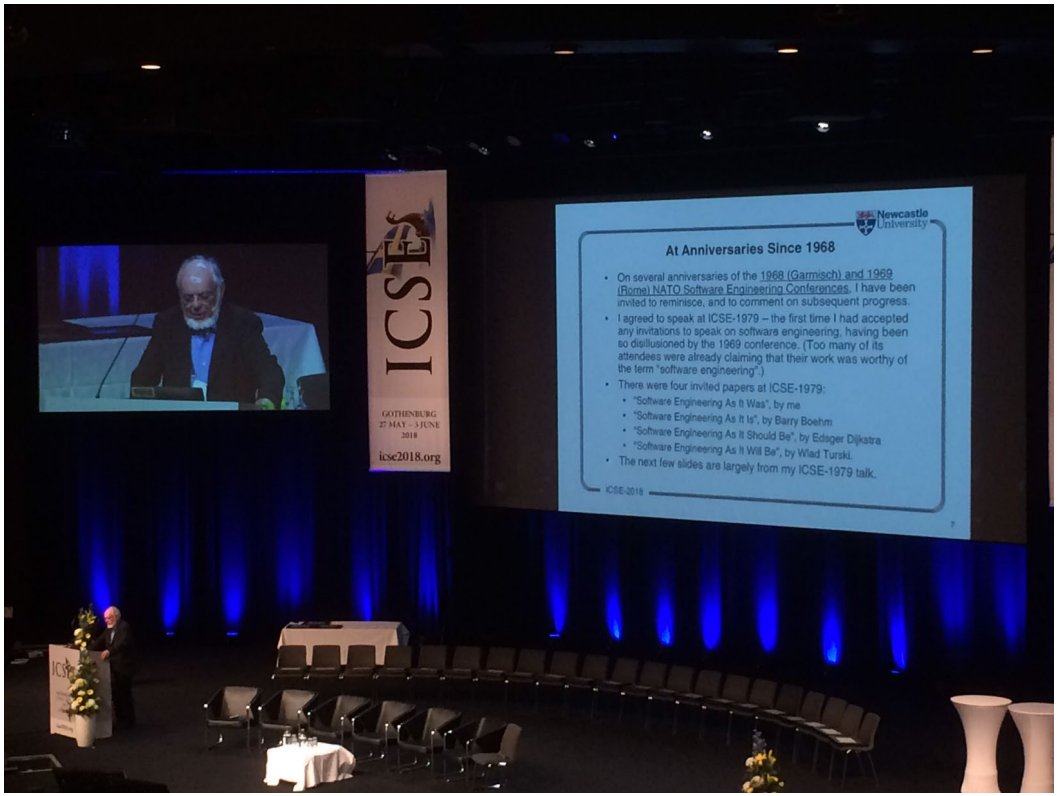
Changes since last year

Software Engineering

- Application of systematic, disciplined, quantifiable approach to software development, operation and maintenance of software. (IEEE-Std.)

The term “Software Engineering” was coined to describe a necessity, not the state of the art.

Freely from Brian Randell, ICSE 2018 Keynote



Agenda:

~~Presentation~~

~~Problems with software~~

~~What is Software Engineering?~~

Course Goals

Labs and Projects (TDDC88/725G64)

Studying Software Engineering

Contents of the theory part

Communication

Examination

Changes since last year

Course goals

Give sound theoretical foundation of Software Engineering

The intended learning outcomes are that the student at the end of the course can:

- explain and exemplify basic concepts in the area of large-scale software engineering.
- explain how to specify, model, implement and test a software system.
- explain how to execute a software development project.

Ambition level

- You will know enough to communicate easily with professional software engineers
- You will have the basic knowledge to start generating your own experience already in student projects
- You will have a curious, but critical, attitude towards existing and new methods

Agenda:

~~Presentation~~

~~Problems with software~~

~~What is Software Engineering?~~

~~Course Goals~~

Labs and Projects (TDDC88/725G64)

Studying Software Engineering

Contents of the theory part

Communication

Examination

Changes since last year

Labs (TDDC88, 725G64 only)

Give the student practical insight in tools used in software development.

- Lab 1 – Packaging with Docker (guided tour)
- Lab 2 – Kubernetes (guided tour)
- Lab 3 - Unified Modeling Language (UML)
- Lab 4 – Automated Testing
- Lab 5 - Software Configuration Management (SCM) and – Continuous integration
- Lab 6 - SW metrics

Sign-up deadline September 7

Projects (TDDC88, 725G64 only)

- TDDC88, 725G64:
- Project integrated in the course during Ht1-Ht2
- Three large “companies” of c:a 25 students
- External customer
- Start: 2024-09-04 at 10:15 in KMC Aulan

- Students of D and U:
- Bachelor project during Vt1-Vt2 2025 for D, 2026 for U
- Teams of 6-8 students
- Real, different customers

Agenda:

~~Presentation~~

~~Problems with software~~

~~What is Software Engineering?~~

~~Course Goals~~

~~Labs and Projects (TDDC88/725G64)~~

Studying Software Engineering

Contents of the theory part

Communication

Examination

Changes since last year

How do you kill this monster?

“There is no silver bullet”

- A good Software Engineer knows several methods, tools, and techniques:
- How does it work?
- What are the benefits?
- What are the drawbacks?
- Is it good for my situation?

Fred Brooks, 1986

"No Silver Bullet — Essence and Accidents of Software Engineering“, *Proceedings of the IFIP Tenth World Computing Conference*.

Software engineering is a craft

STUDYING IS IMPORTANT, BUT…

… it also needs to be practiced!

THERE ARE VERY FEW SIMPLE ANSWERS

The answer to just about everything is ”it depends”.

There are always pros and cons.

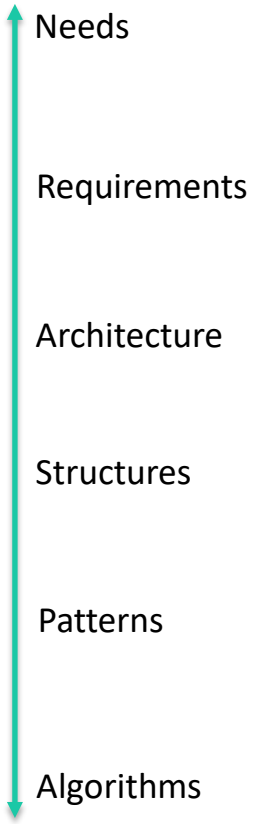
There is a lot of buzz.

THE MOST IMPORTANT SKILL IS TO THINK

Think creatively.

Think critically.

Move comfortably between levels of abstraction.



How does one study software engineering?

READ MULTIPLE SOURCES

There is always another side to the latest trend. Look for skeptical voices.

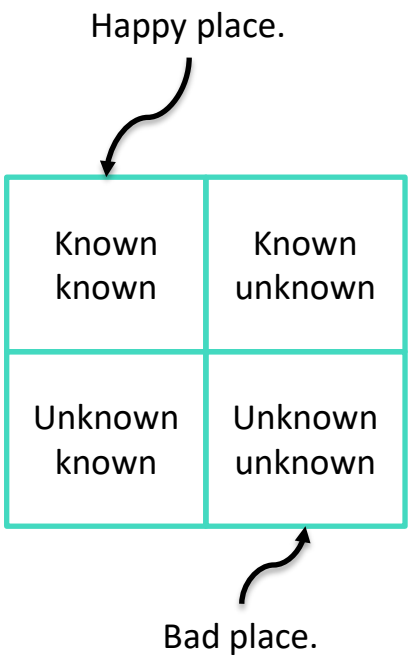
TRUST EXPERIENCE OVER THEORY

There are many nice theoretical concepts that don't work in practice.

The world is full of clever languages and compilers that nobody ever uses.

YOU WILL NEVER LEARN EVERYTHING

Don't even try. But it helps to know what you don't know!



Example

One approach:

- Check the “concepts”-file
- Google
- Wikipedia
- References
- Search and read

Another approach:

- Homepage
- Literature
- Download and read

Agenda:

~~Presentation~~

~~Problems with software~~

~~What is Software Engineering?~~

~~Course Goals~~

~~Labs and Projects (TDDC88/725G64)~~

~~Studying Software Engineering~~

Contents of the theory part

Communication

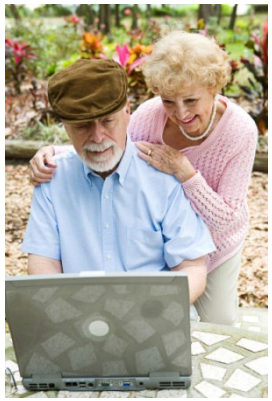
Examination

Changes since last year

Scope of a software life-cycle model

Idea

Software Product

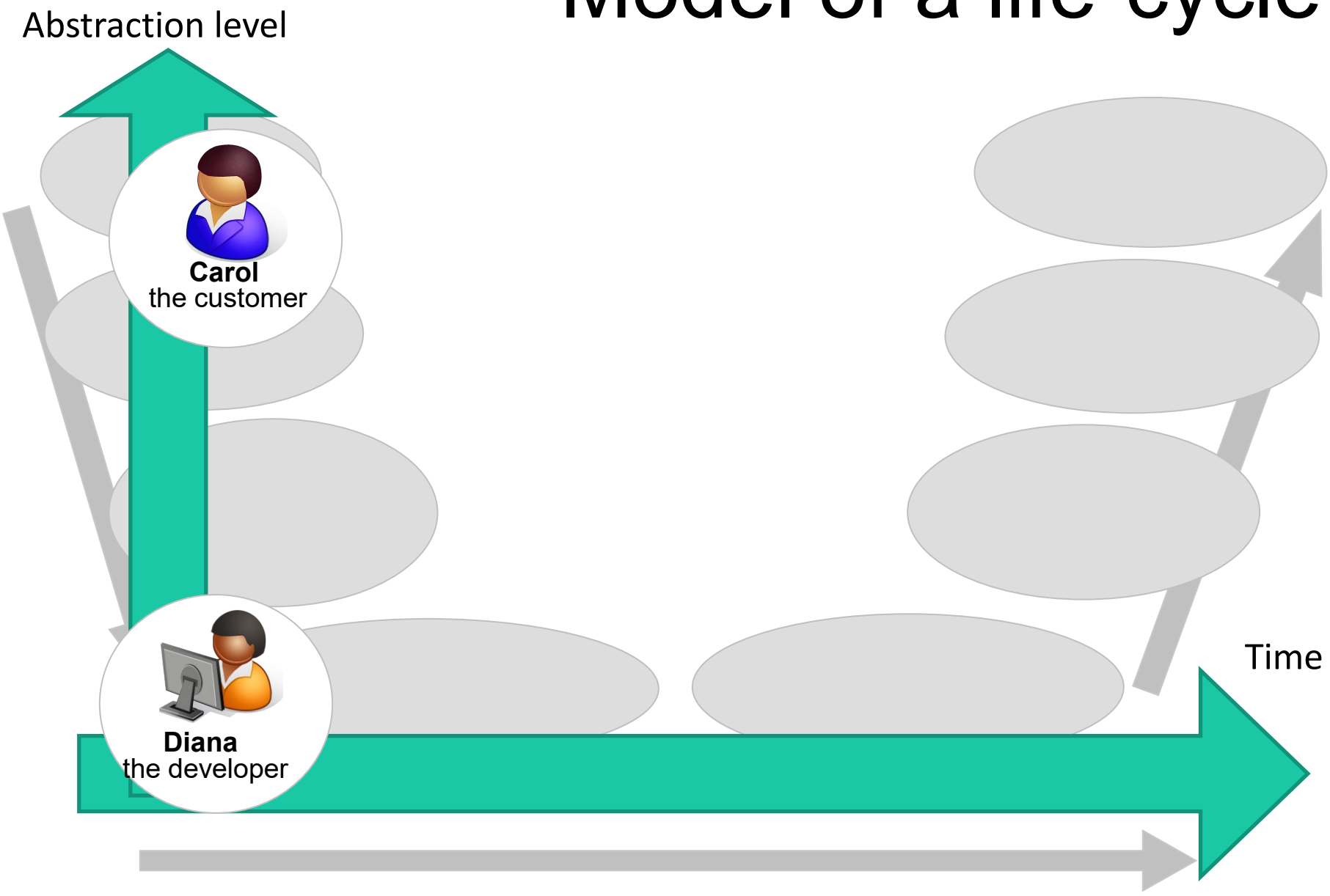


Usage

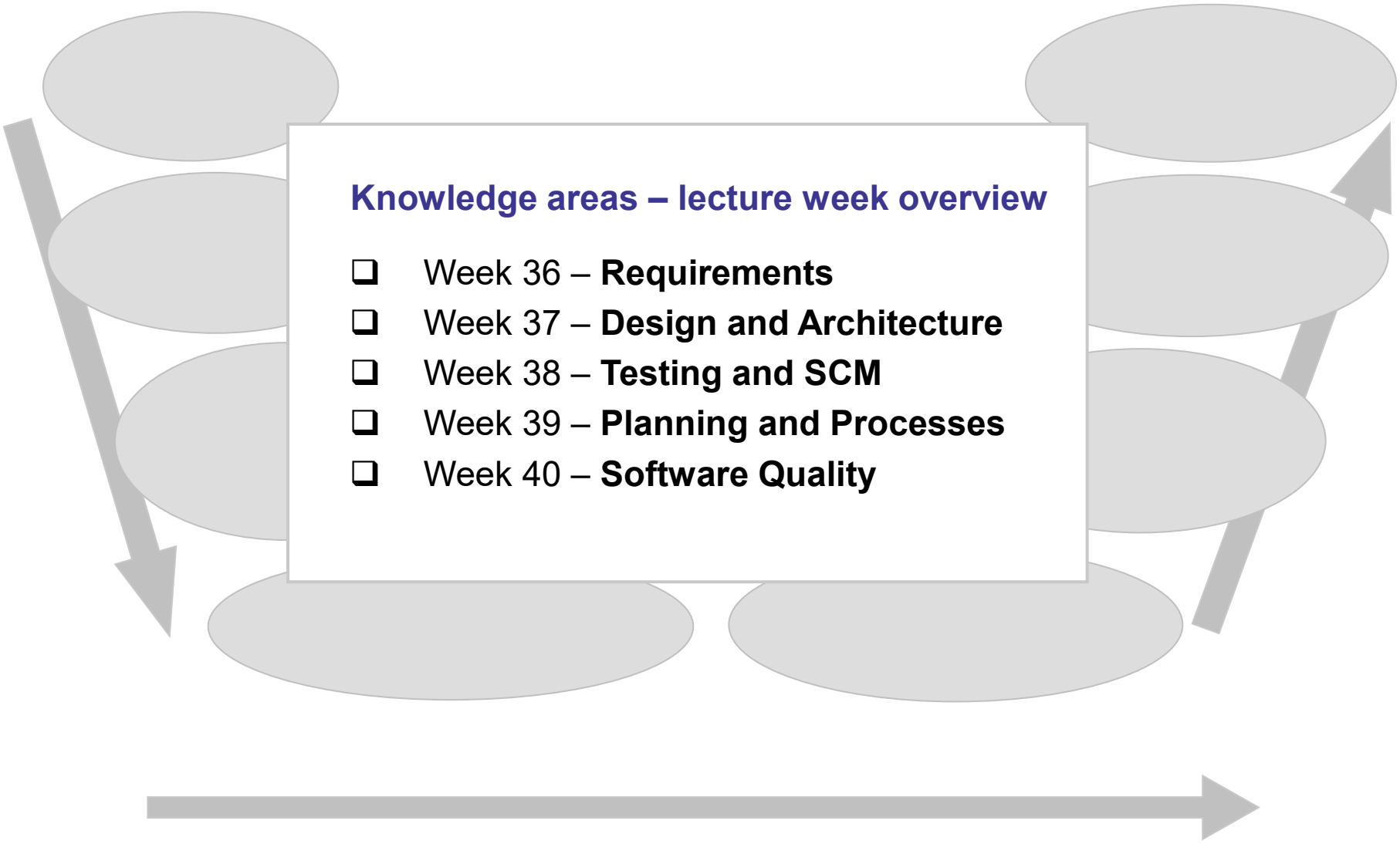
Operation & maintenance

Replacement

Model of a life-cycle



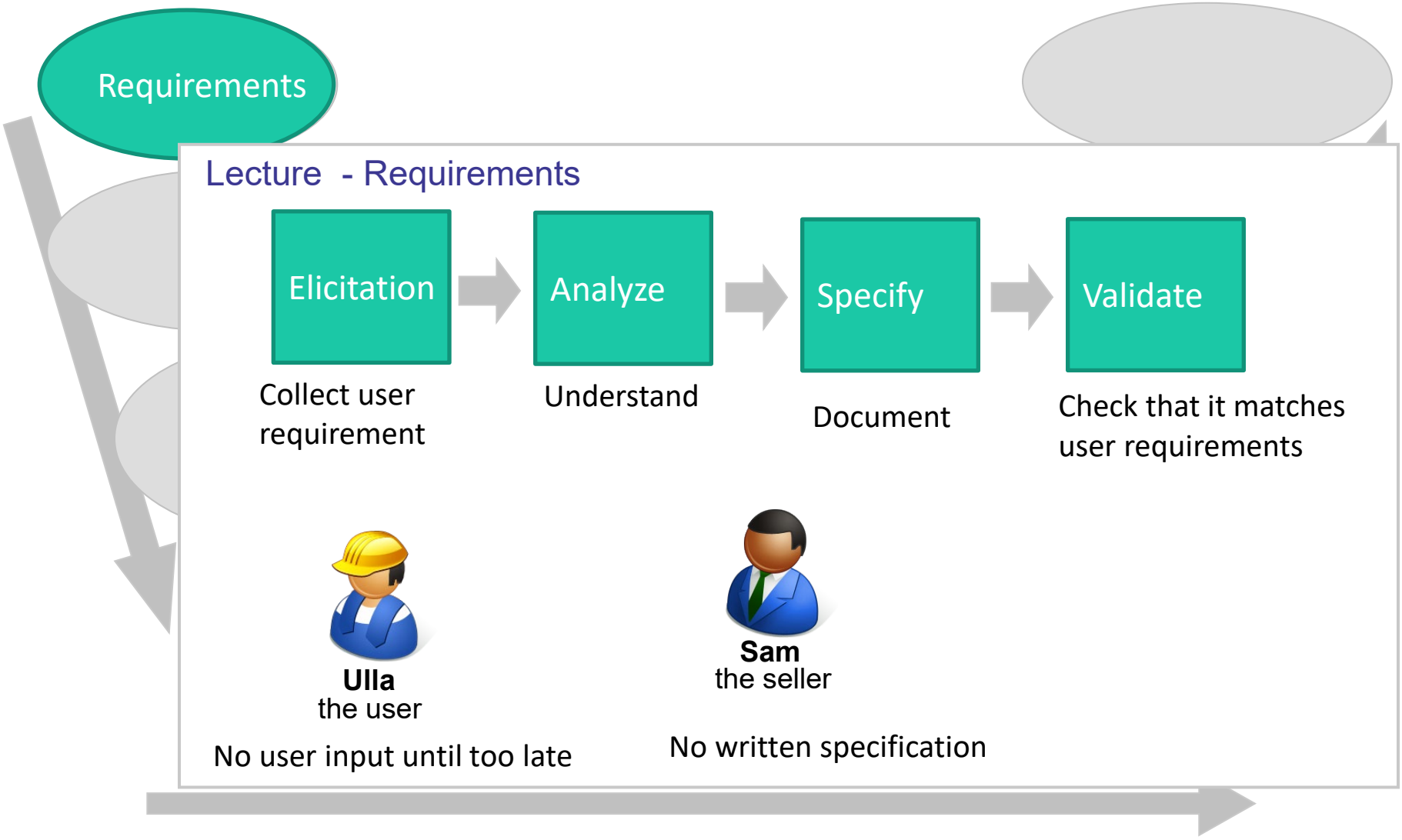
Model of a life-cycle



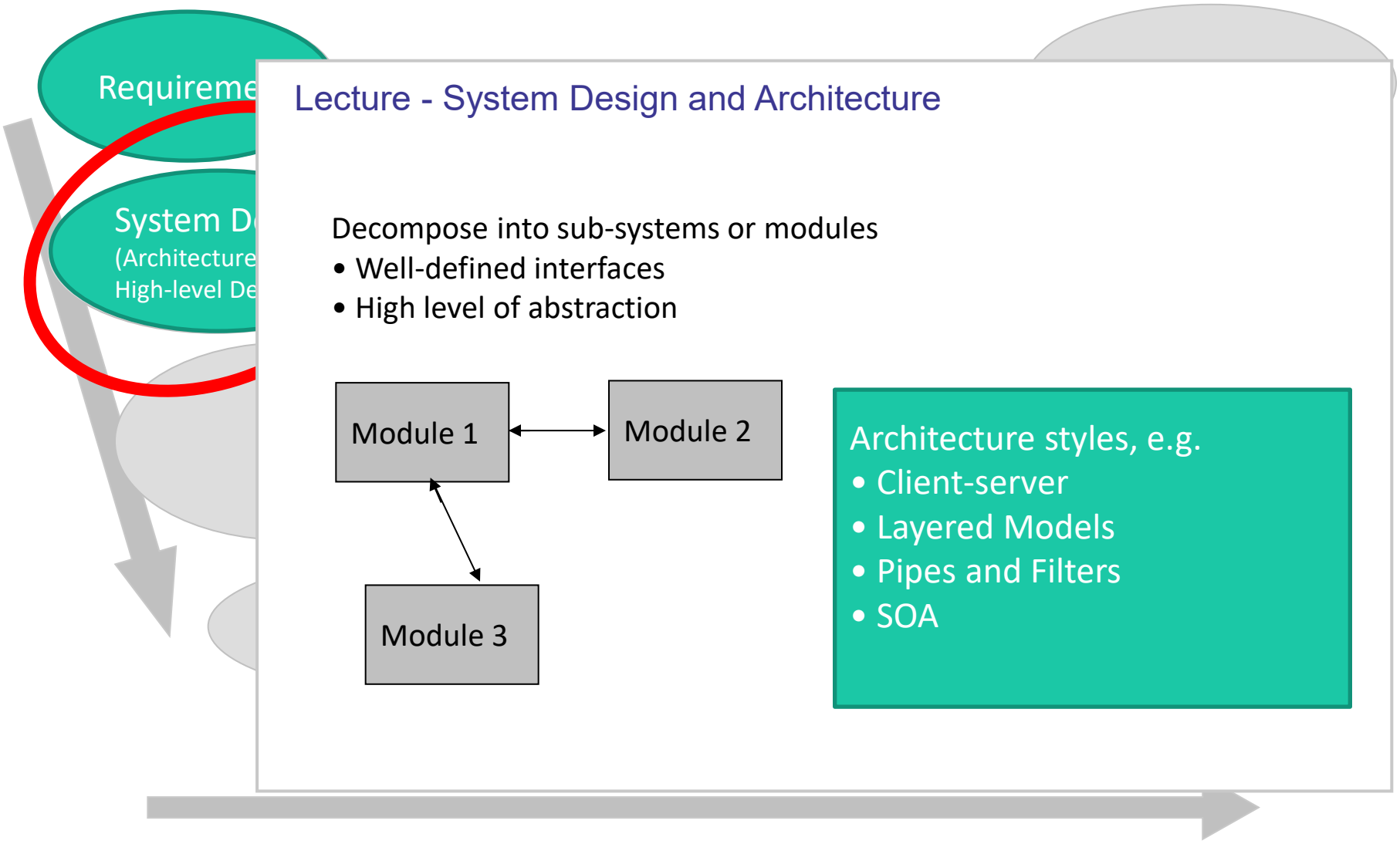
Knowledge areas – lecture week overview

- Week 36 – **Requirements**
- Week 37 – **Design and Architecture**
- Week 38 – **Testing and SCM**
- Week 39 – **Planning and Processes**
- Week 40 – **Software Quality**

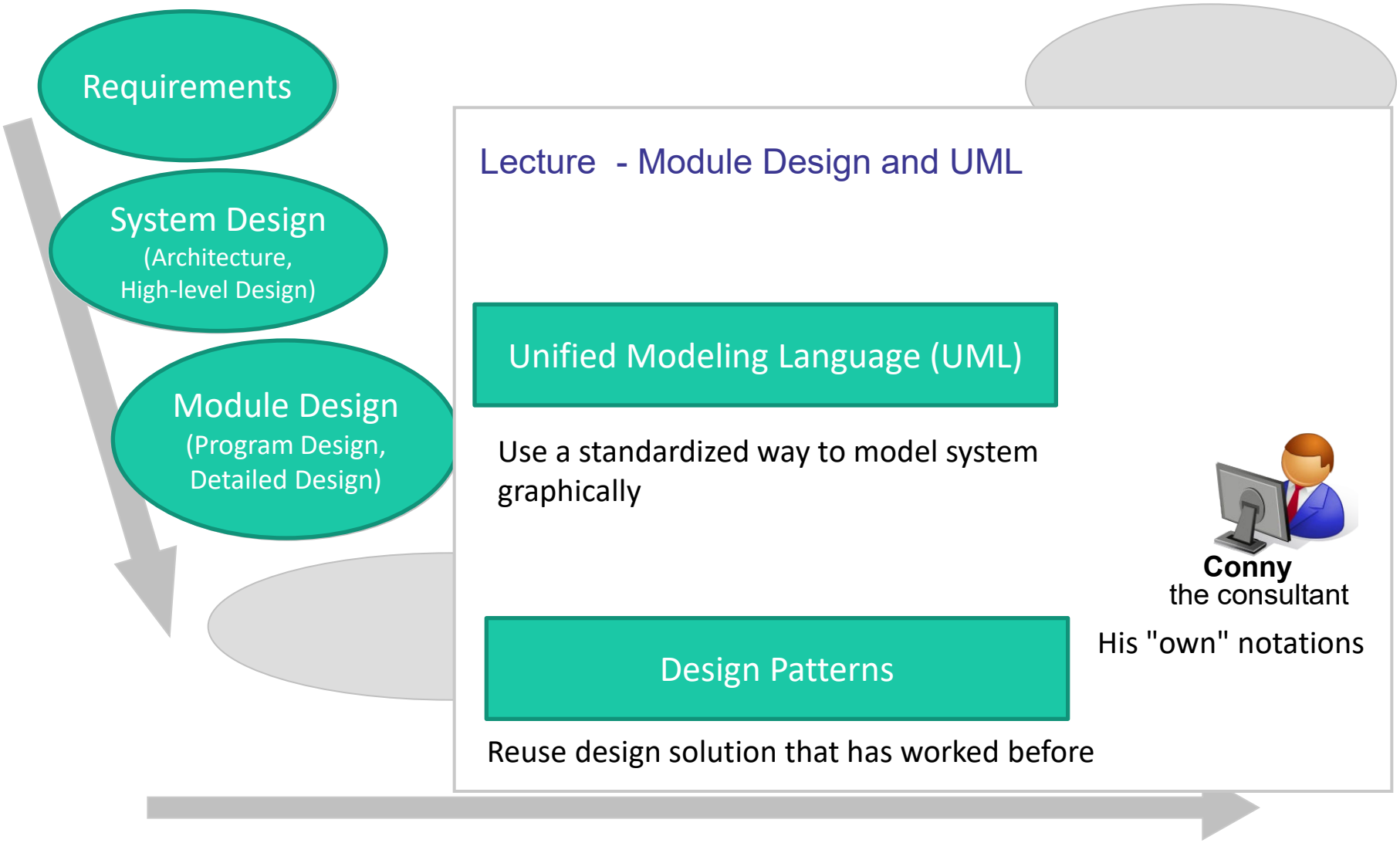
KA #1: Requirements



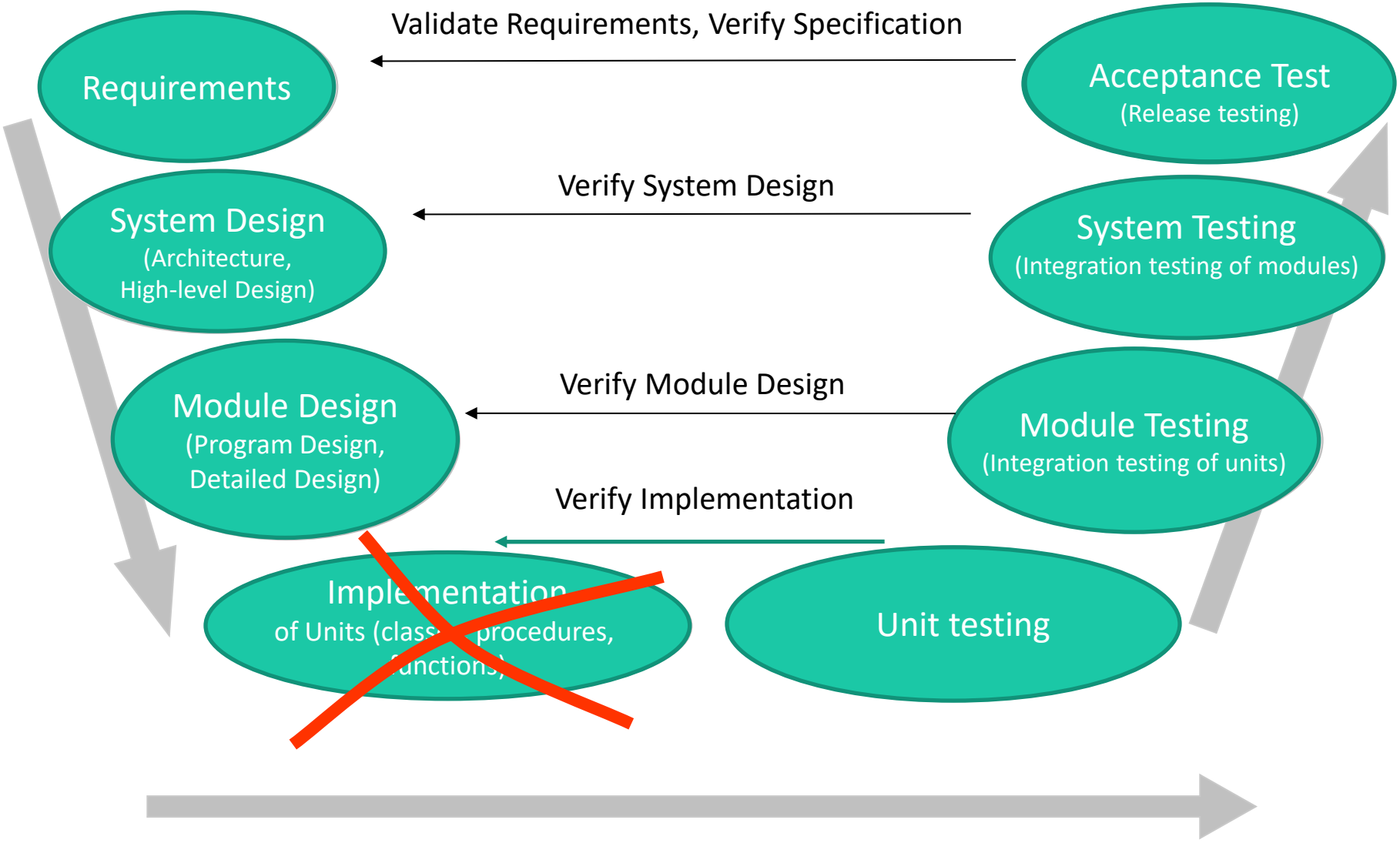
KA #2: Design and Architecture



KA #2: Design and Architecture



KA #3: Testing and SCM



KA #3: Testing and SCM

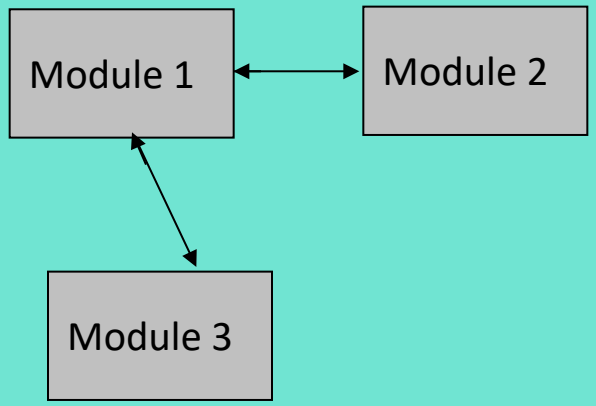
Validate Requirements, Verify Specification

Requirements

Acceptance Test
(Release testing)

Sys
(A
High

Lecture - Testing

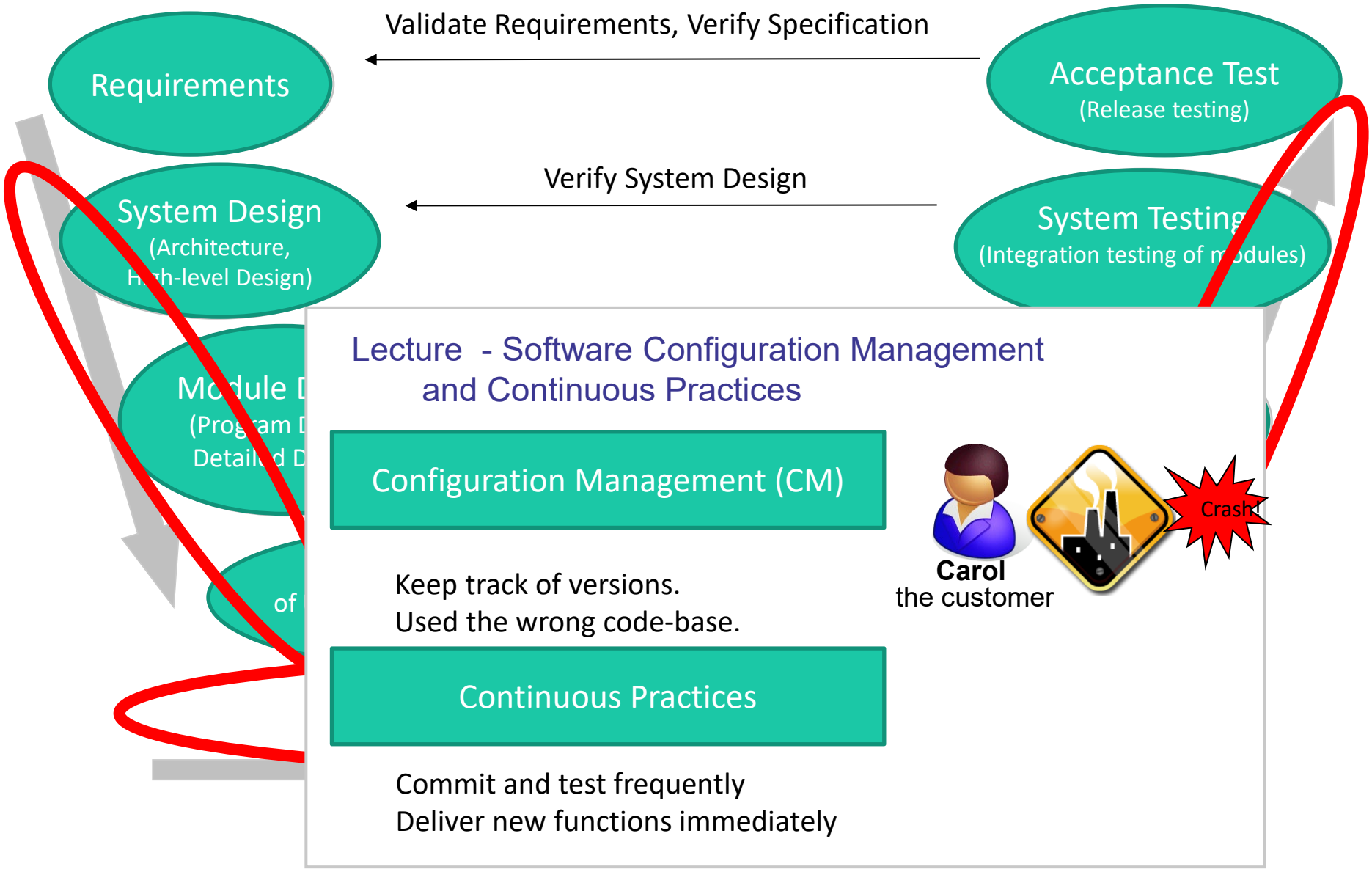


Integration testing
- Dependencies between modules

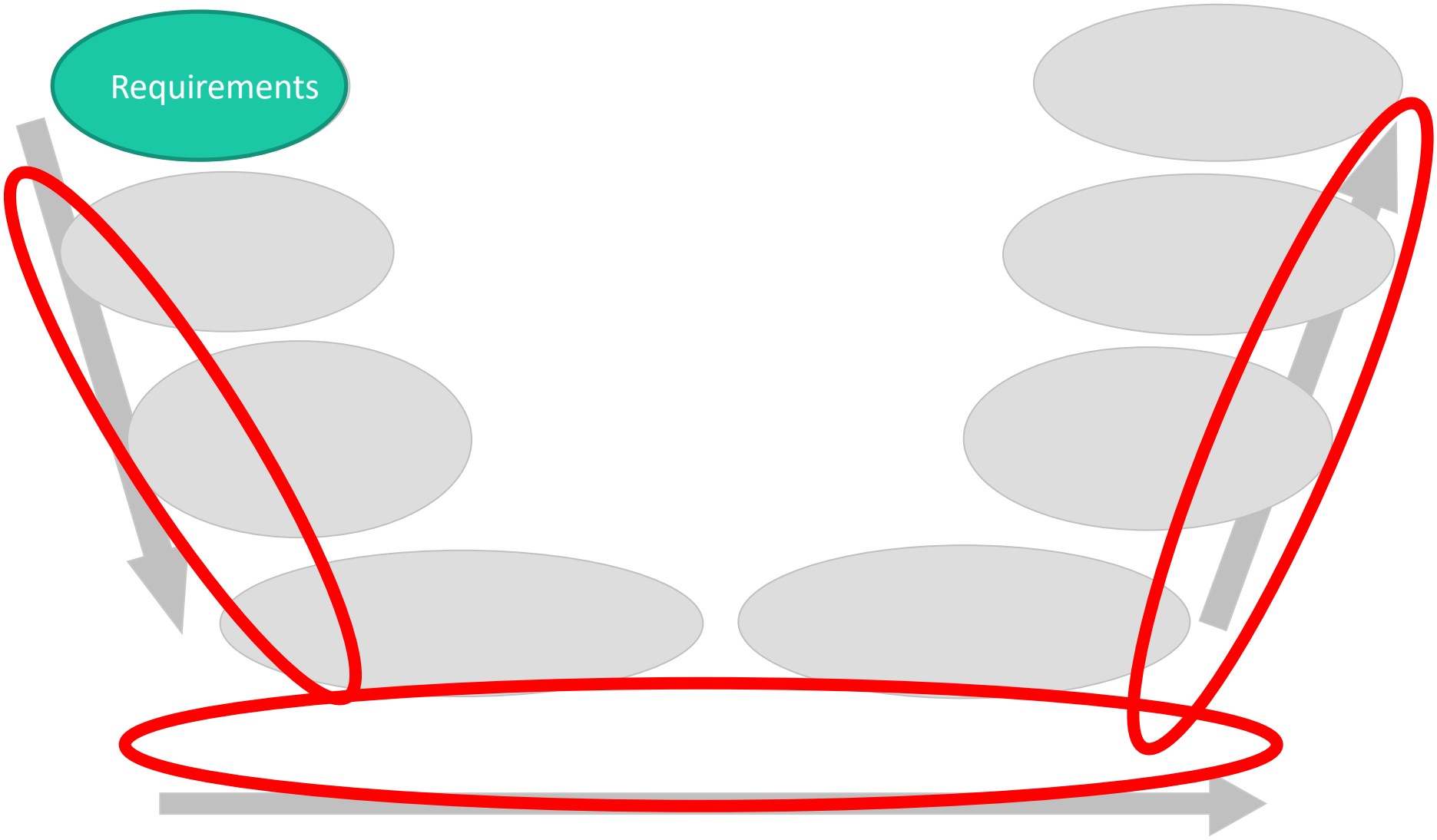
Approaches
- Top-down (need stubs)
- Bottom-up (need drivers)
- Sandwich
- Big-bang

Lack of unit, integration and system testing

KA#3: Testing and SCM



Supporting Processes



KA #4: Planning and Processes

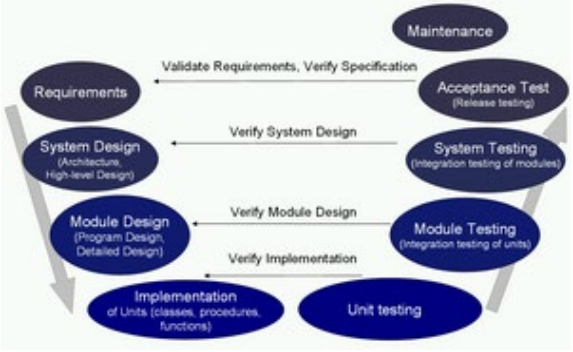
Requirements

Lecture - Software Life Cycles and Processes

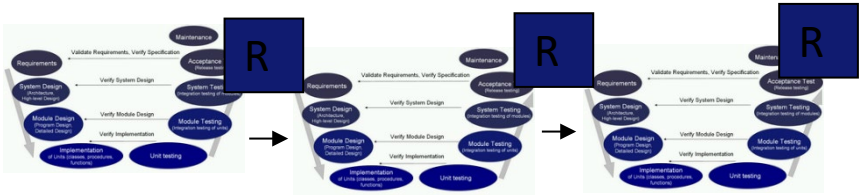
Agile Methodologies

V-model

Waterfall model



Iterative models



Processes e.g. Scrum, XP, OpenUP, Kanban, Essence

KA #4: Planning and Processes

Lecture - Project Management

Resources

Calendar-time

Time plan

Milestones

Time-budget

Activities

- Resource management - Harry
- Time-buffer, toll-gates
- Internal milestones - keep track

Harry
the hacker

Risk Management

Identify

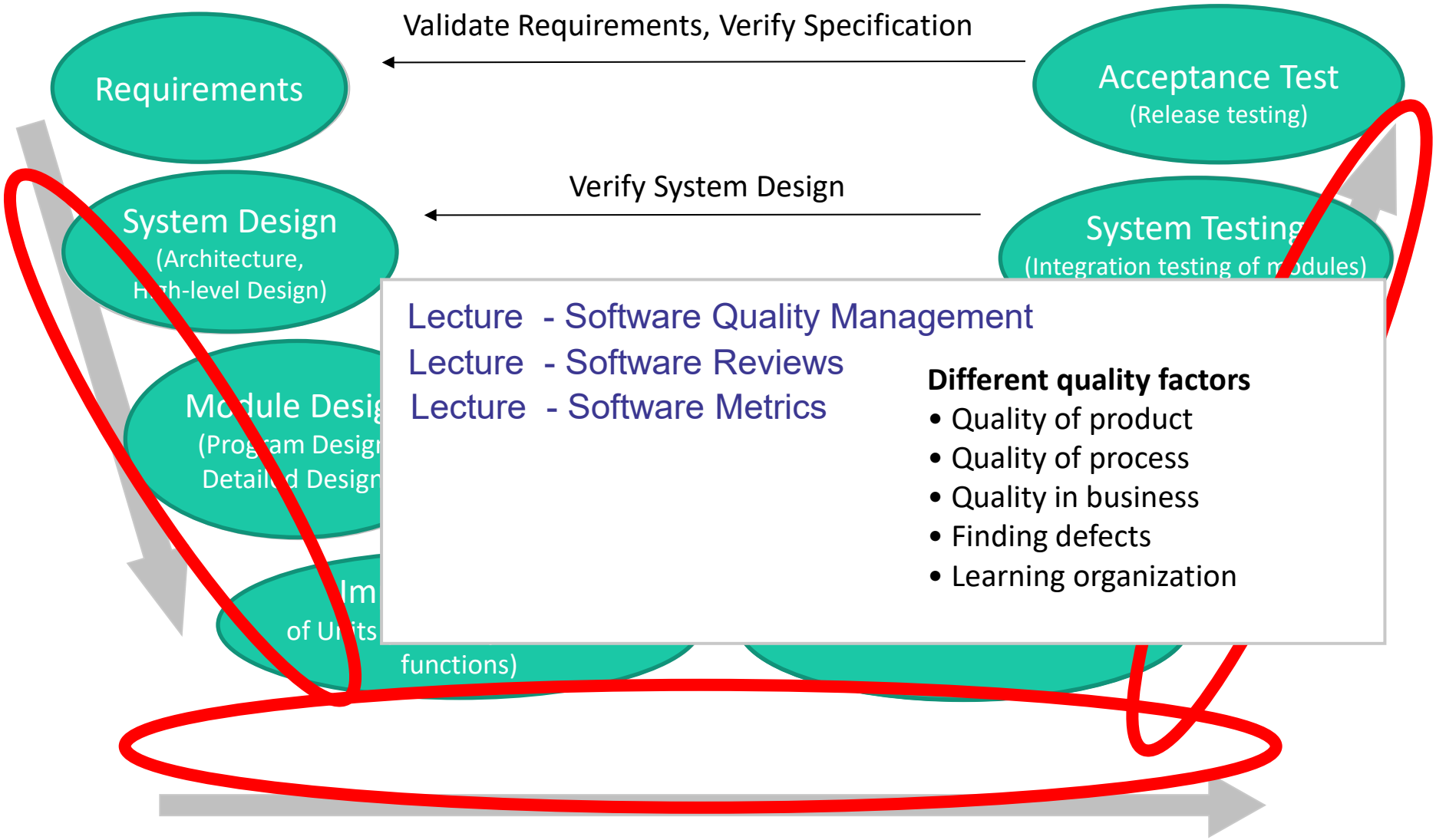
Analyze

Plan

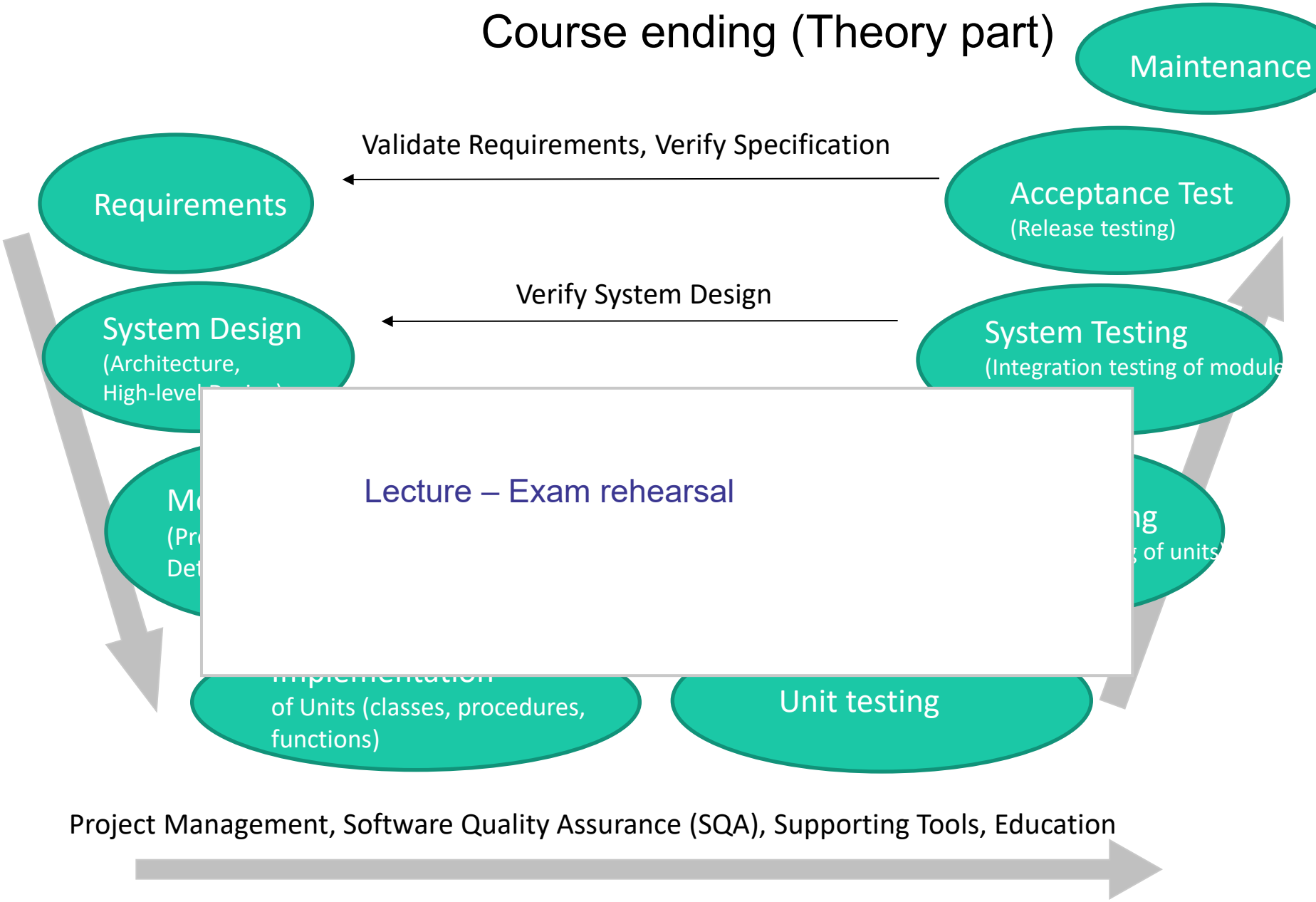
- Minimize
- Transfer
- Accept

probability - consequence

KA #5: Software Quality

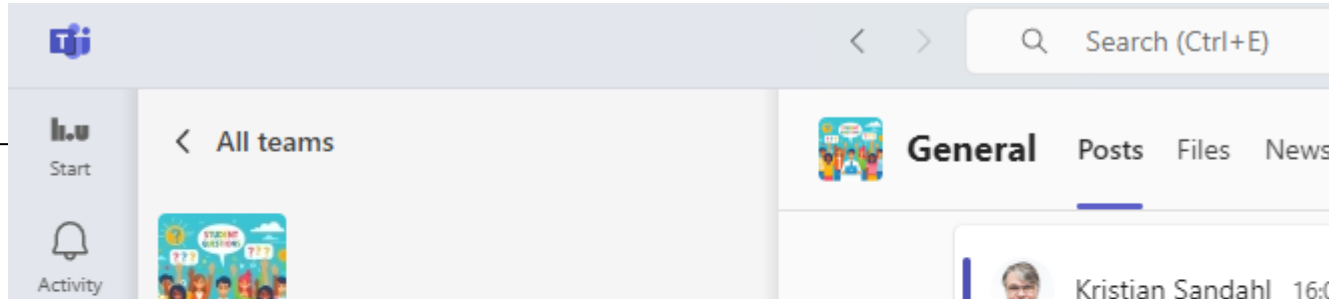


Course ending (Theory part)



Communication Principles

- What if "*I have a question related to the course*"?
  - **Principle 1**: Post your technical questions in the MS Teams channels of the course
  - (You may get no answer if you write an email instead)
- What if "*I have a personal issue / problem*"?
  - **Principle 2**: Write an email to Kristian or DanielV if you have a personal issue or problem in the course
  - (You may get too many replies if you write a post instead)



Agenda:

~~Presentation~~

~~Problems with software~~

~~What is Software Engineering?~~

~~Course Goals~~

~~Labs and Projects (TDDC88/725G64)~~

~~Studying Software Engineering~~

~~Contents of the theory part~~

Communication

Examination

Changes since last year

Feed-back

Muddy Cards, Wednesday, week 37

All feedback is welcome

- Via mail kristian.sandahl@liu.se, daniel.varro@liu.se
- Via phone 0706-681957
- Via coffee. Drop an e-mail, so we can schedule a time

EVALIUATE (web-based system)

Student representative feed-back

INCLUDE COURSE CODE IN MAIL HEADINGS

Agenda:

~~Presentation~~

~~Problems with software~~

~~What is Software Engineering?~~

~~Course Goals~~

~~Labs and Projects (TDDC88/725G64)~~

~~Studying Software Engineering~~

~~Contents of the theory part~~

~~Communication~~

Examination

Changes since last year

Examination

Course part	Examination	Credits	Applicable to
Theory	Written exam	4 hp	TDDC93, TDDC88, 725G64
Project	Project tasks	6 hp	TDDC88, 725G64
Laboratory exercises	Oral exam, written exercises	2 hp	TDDC88, 725G64

Written Exam (both TDDC88/725G64 and TDDC93)

- Primary Exam 14:00-18:00, October 28, 2024
- Retake Exam January 8, 2025
- Retake Exam, August 2025, TBD

Examination

Pass with 50 credits

Part I: Fundamentals

- Requirements
- Planning and Processes
- Design and Architecture
- Testing and SCM
- Software Quality

10 credits per area. Max 50 credits.

Part II: Advanced

50 credits, distributed over 2-5 questions.

- argue, compare, and analyze different concepts and techniques.
- construct and/or design solutions to larger problem.
- explain more advanced and specific topics.

Lecture exercises

Lecture Exercises

- Optional
- 5 exercises, one for each knowledge area
- Solve exercises in groups of 2 students
- 0-4 extra credits on the exam for each exercise (max 20)
- Possibility: Higher grades and easier to pass the exam

Note! The credits are **only** valid on the exams in October 2024, January 2025, and August 2025

For details, see course page under "Lecture Exercises"

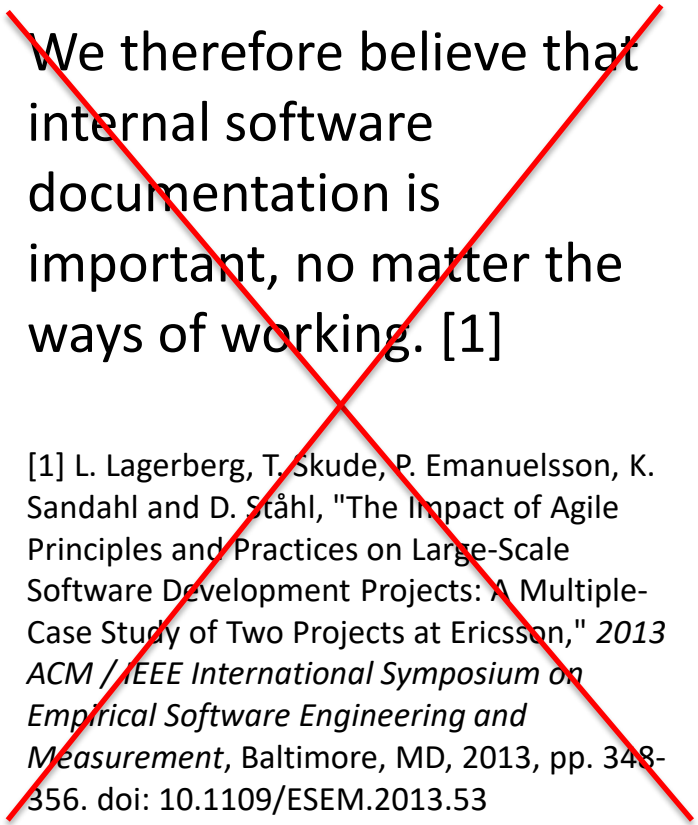
and lab assignments

Lecture exercise: Plagiarism is forbidden

https://liu.se/en/article/plagiering-upphovsratt

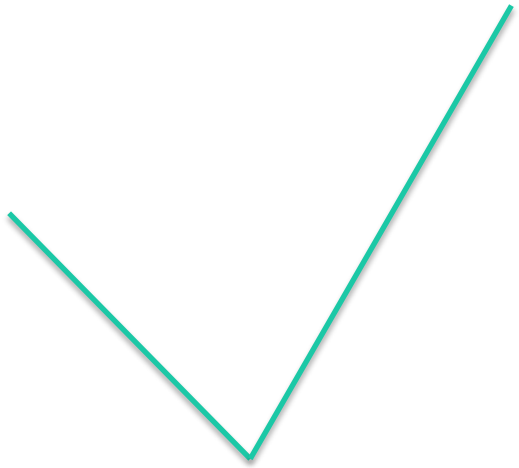
"We therefore believe that internal software documentation is important, no matter the ways of working." [1]

[1] L. Lagerberg, T. Skude, P. Emanuelsson, K. Sandahl and D. Ståhl, "The Impact of Agile Principles and Practices on Large-Scale Software Development Projects: A Multiple-Case Study of Two Projects at Ericsson," *2013 ACM / IEEE International Symposium on Empirical Software Engineering and Measurement*, Baltimore, MD, 2013, pp. 348-356. doi: 10.1109/ESEM.2013.53



Lecture exercise: Run NoPlagiat self-study

A large-scale study found that regardless of working method, internal documentation is important. [1]



[1] L. Lagerberg, T. Skude, P. Emanuelsson, K. Sandahl and D. Ståhl, "The Impact of Agile Principles and Practices on Large-Scale Software Development Projects: A Multiple-Case Study of Two Projects at Ericsson," *2013 ACM / IEEE International Symposium on Empirical Software Engineering and Measurement*, Baltimore, MD, 2013, pp. 348-356. doi: 10.1109/ESEM.2013.53

Don’t use generative AI to produce text and code that you should have written yourself!

Applies to: Lecture exercises, individual project reports, solutions to labs, written assignments

Agenda:

~~Presentation~~

~~Problems with software~~

~~What is Software Engineering?~~

~~Course Goals~~

~~Labs and Projects (TDDC88/725G64)~~

~~Studying Software Engineering~~

~~Contents of the theory part~~

~~Communication~~

~~Examination~~

Changes since last year

Changes since last year

- ISO25010 material updated to 2023 version
- New project, new customer
- Lecture exercises better aligned with exam
- Slightly modified lecture exercises. Students from previous years must have at least 30% new material in answers.
- More examples and best practices for UML modeling
- Re-take exams in 2025 will change from Wiseflow to Inspera

Summary

- SE is critical for software industry
- Be skeptical and critical to methods
- Knowledge areas:
  - Requirements
  - Design and architecture
  - Testing and SCM
  - Planning and processes
  - Software quality
- Communication in Teams
- Examination

Extra material

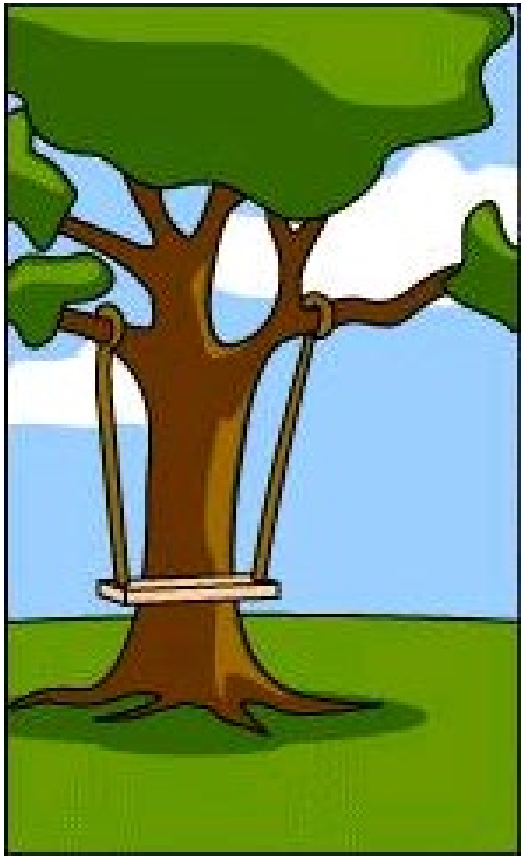
www.liu.se

LINKÖPING UNIVERSITY

A typical Software Project

How the customer explained it

A typical Software Project

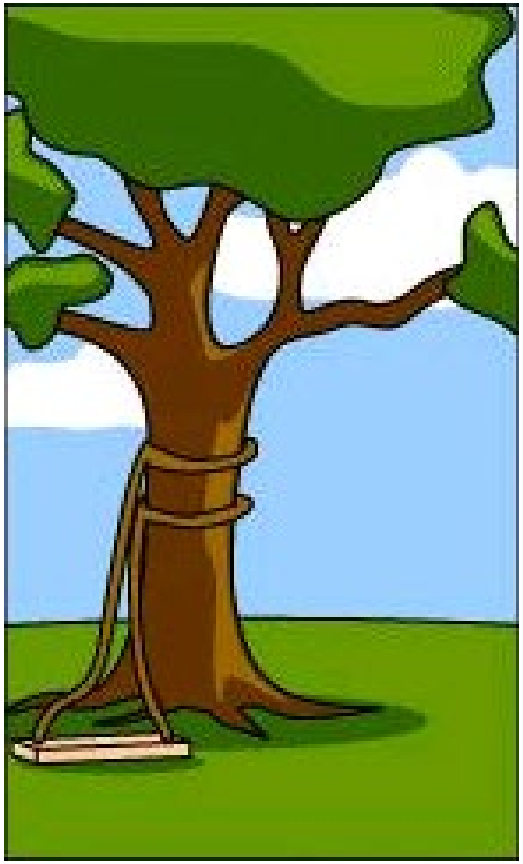


How the project leader understood it

A typical Software Project

How the analyst designed it

A typical Software Project

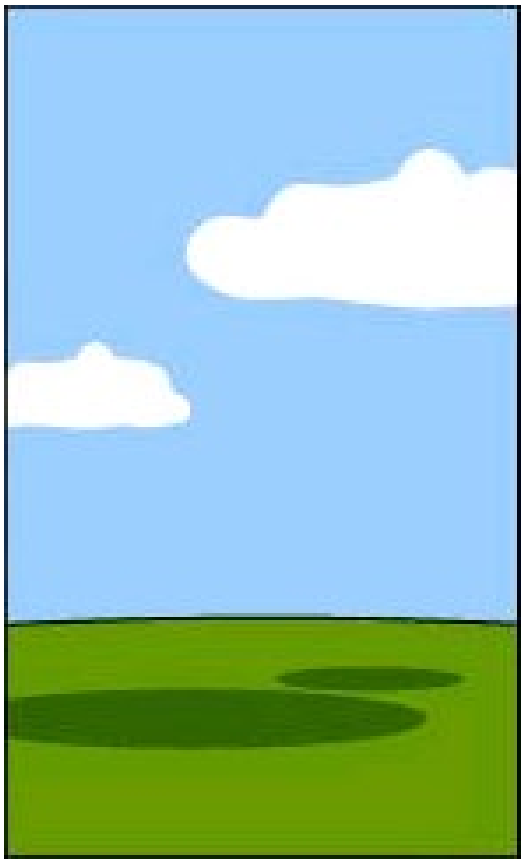


How the programmer wrote it

A typical Software Project

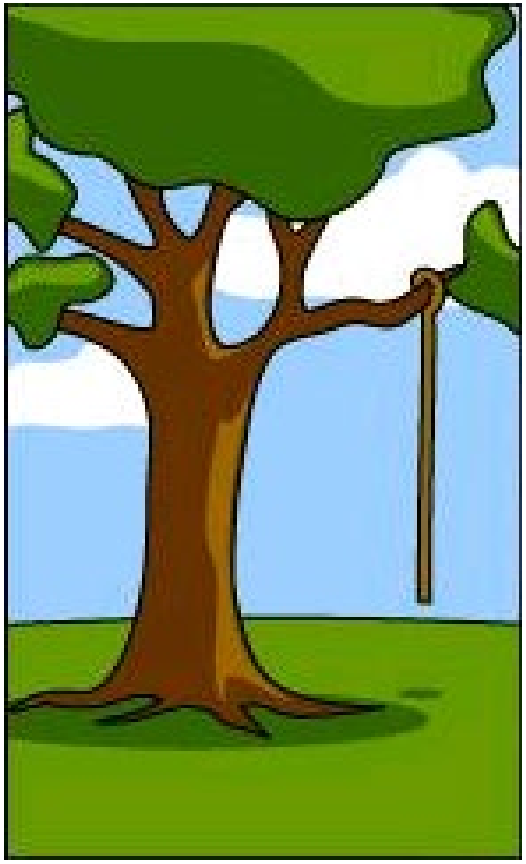
How the business consultant described it

A typical Software Project



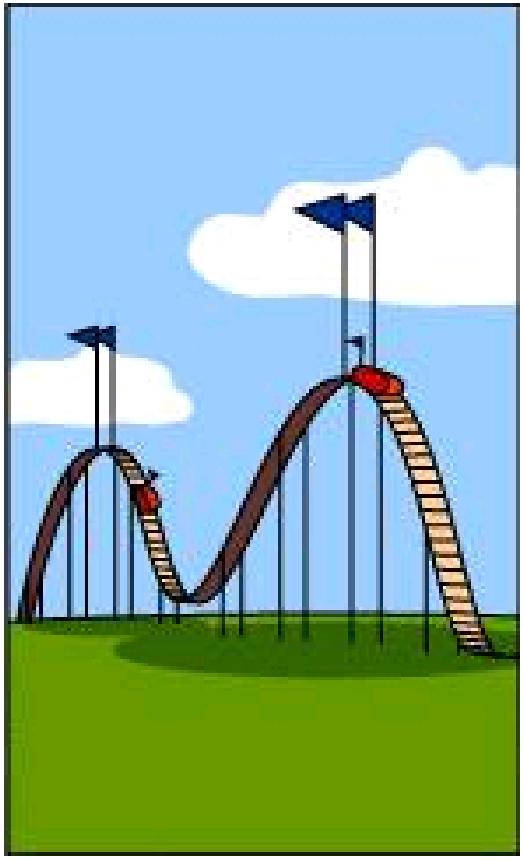
How the project was documented

A typical Software Project



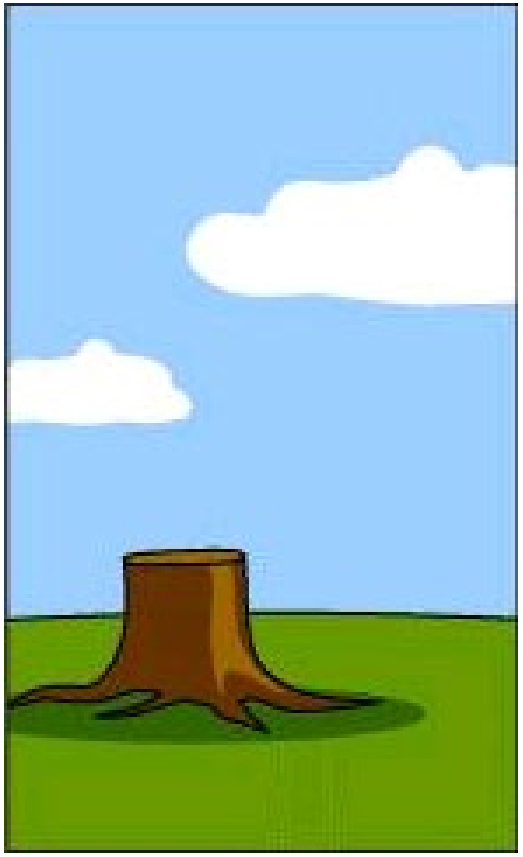
What operations installed

A typical Software Project



How the customer was billed

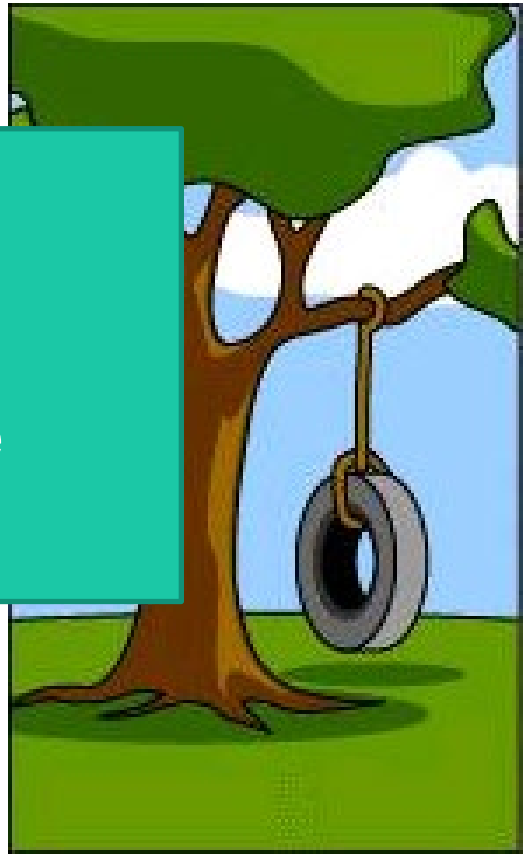
A typical Software Project



How it was supported

A typical Software Project

Welcome to
the Software
Engineering course
2024!



Finally, what the customer really needed