Software Engineering

1. Introduction | Matthias Tichy, Sabrina Böhm, Timo Zuccarello | October 15, 2024

SoftVarE

Software Engineering
Programming Languages

universität ulm

Lecture Overview – 1. Introduction

1. Course Introduction
2. What is Software (Engineering)?
3. How Relevant is Software?

1. Introduction

About The Lecturers

Prof. Dr. Matthias Tichy

- 1997–2002: studied Computer Science
- 2002–2009: PhD student in Paderborn
- 2009–2012: PostDoc in Paderborn
- 2010–2011 Acting Professor in Augsburg
- 2012–2015: Assistant / Associate Professor in Gothenburg (Schweden)
- 2015–now: Professor in the Institute of Software Engineering and Programming Languages

About The Lecturers

M.Sc. Sabrina Böhm

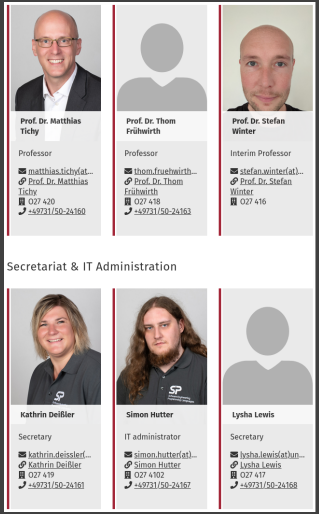
- head of Sopra
- 2016–2022: studied Computer Science
- 2022–now: PhD student in the Institute of Software Engineering and Programming Languages

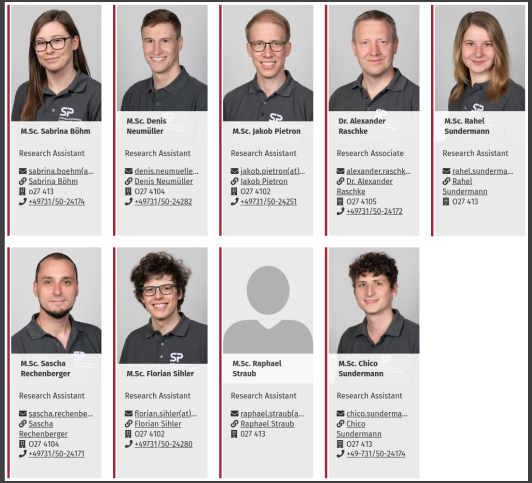
About The Lecturers

B.Sc. Timo Zuccarello

- Exercises
- 2016–now: studied Computer Science

Institute of Software Engineering and Programming Languages





Institute of Software Engineering and Programming Languages

Software Engineering

How to develop and operate software?

Research Areas

- Software Configuration
- Software Evolution
- Developer Experience
- Quality Assurance

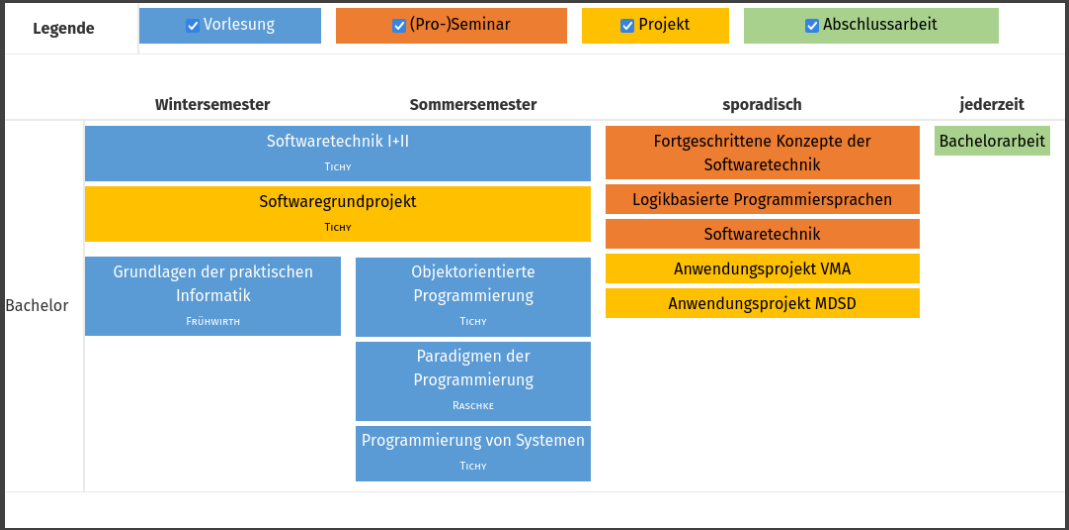
Programming Languages

How to communicate with machines?

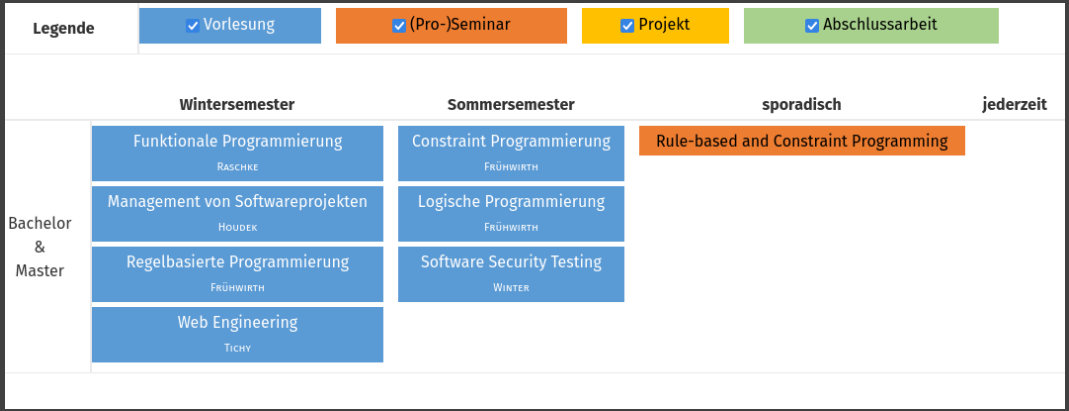
Research Areas

- Domain-Specific Languages
- Declarative and Functional Programming

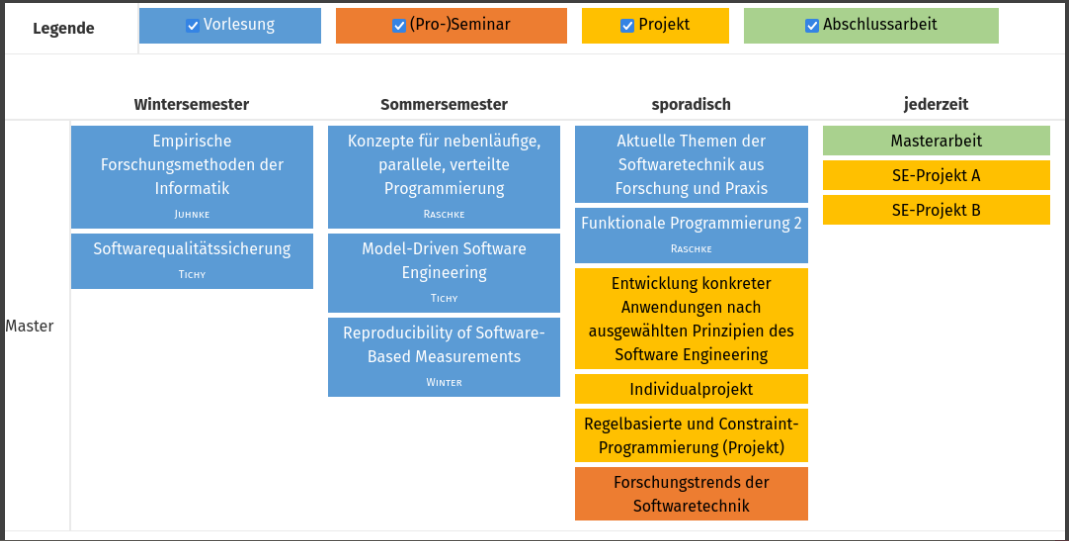
Bachelor Courses of Our Institute



Bachelor and Master Courses of Our Institute



Master Courses of Our Institute



Current Courses of Our Institute

Bachelor

- Fundamentals of Practical Computer Science (Grundlagen der praktischen Informatik)
- Object-Oriented Programming (Objektorientierte Programmierung)
- Software Engineering I+II (Softwaretechnik I+II)
- Software Project (Softwaregrundprojekt)
- Proseminar on Software Engineering (Softwaretechnik)

Bachelor and Master

- Functional Programming (Funktionale Programmierung)
- Management of Software Projects (Management von Softwareprojekten)
- Rule-Based Programming (Regelbasierte Programmierung)
- Software Quality Assurance (Softwarequalitätssicherung)
- Web Engineering
- Seminar on Rule-Based and Constraint Programming (Regelbasierte und Constraint-Programmierung)
- Numerous Projects

Master

- Model-Driven Software Engineering (Modellgetriebene Softwareentwicklung)
- Empirical Research Methods in Computer Science (Empirische Forschungsmethoden der Informatik)

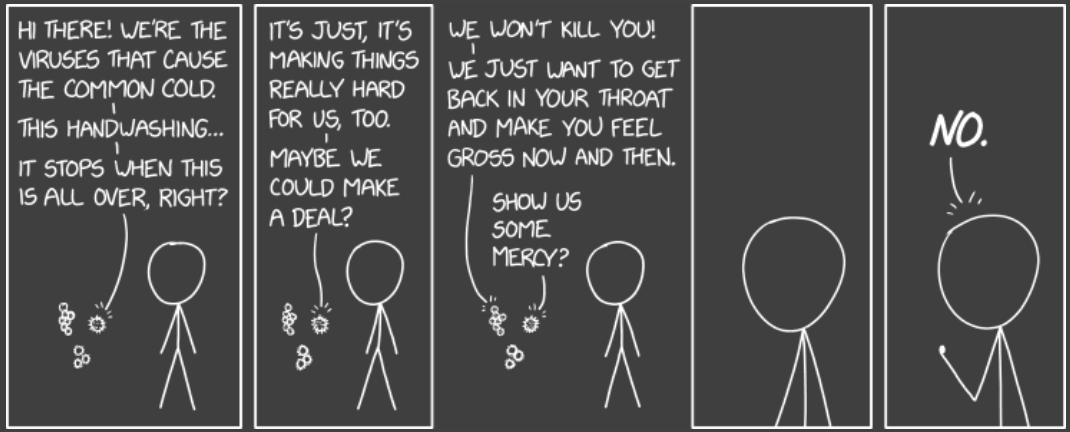
About This Course

Software Engineering

- German title: Softwaretechnik
- Former German title: Softwaretechnik I + II
- Abbreviation: SWT
- Credits: 6 ECTS (60h+120h=180h)
- Semester hours: 4
- Requirements (contentual): Object-Oriented Programming
- Courses of studies:
  - B.Sc.: Informatik, Medieninformatik, Software Engineering, Informationssystemtechnik
  - M.Ed.: Informatik Lehramt
  - ...

Course Organization

- Lecture over winter ~~and summer~~ term
- Up-to two lectures per week, see schedule in Moodle
- every two weeks exercise (formal exercises) on Wednesday (Timo Zuccarello)
- Written exam at the end of the winter term
- In parallel: Software Project (Softwareprojekt) (5 + 5 = 10 ECTS) (Sabrina Böhm)


HI THERE! WE'RE THE VIRUSES THAT CAUSE THE COMMON COLD.
THIS HANDWASHING... IT STOPS WHEN THIS IS ALL OVER, RIGHT?
IT'S JUST, IT'S MAKING THINGS REALLY HARD FOR US, TOO.
MAYBE WE COULD MAKE A DEAL?
WE WON'T KILL YOU!
WE JUST WANT TO GET BACK IN YOUR THROAT AND MAKE YOU FEEL GROSS NOW AND THEN.
SHOW US SOME MERCY?
NO.

Lecture and Videos

Hybrid-Format

- on-site lecture with interactive elements
- extensive use of Moodle:
  small tasks, your questions, your answers
- use tutorial groups of software project for questions

What if I cannot attend the lecture?

- videos available in Moodle and on Youtube (from 2020/2021)
- exception: the last lecture
- use slides in parallel to check for new and changed content
- watch videos together!
- do interactive parts together!

Literature for This Course

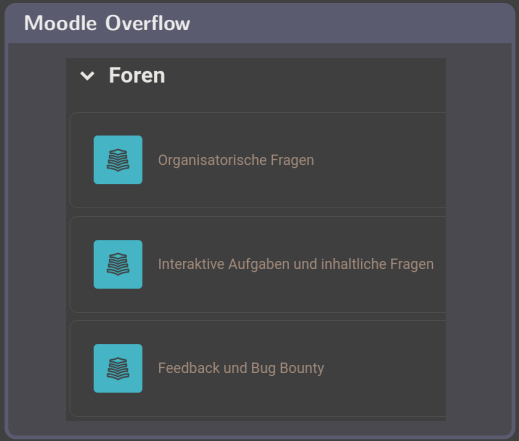
[Sommerville]

- Ian Sommerville. Software Engineering, 10. Edition, Pearson, 2018.
  - German, English, and earlier versions
  - Videos by Ian Sommerville and others available online
- More literature announced in each lecture

Asking Questions

How to Find Answers

1. Ask questions during the lecture (e.g., during interactive parts)
2. Check information in Moodle Overflow
   - Check already answered questions in Moodle overflow
   - Ask your own questions and answer questions of fellow students
3. Ask questions in your tutorial group of the software project (Sopra)
4. Meet me: on demand (and typically in my office O27-419)
5. Contact me via matthias.tichy@uni-ulm.de



Course Introduction

Lessons Learned

- Who are we?
- How is this course organized?

Practice

Ask questions in lecture or Moodle, if any

1. Introduction

1. Course Introduction

2. What is Software (Engineering)?

Characteristics of Software

Software (Products)

Application and System Software

Properties of Software

Software Engineering

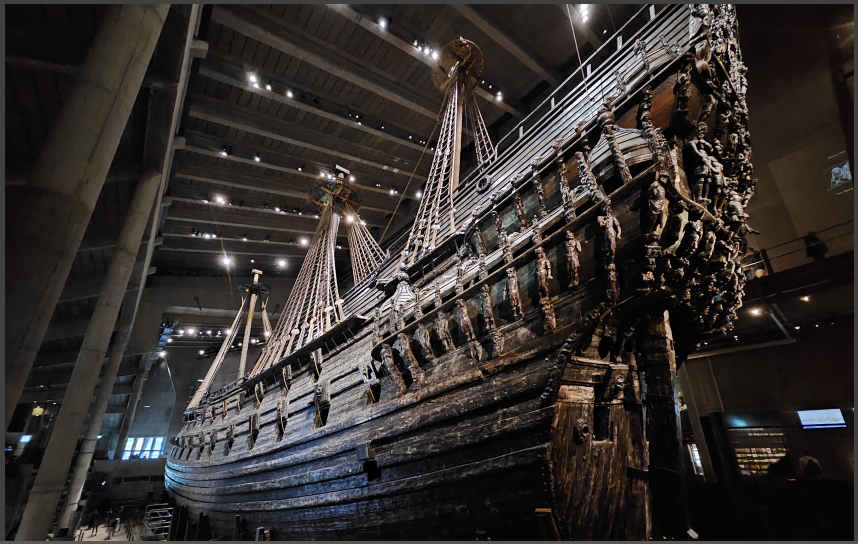
Software and System Engineering

(Software) Engineering

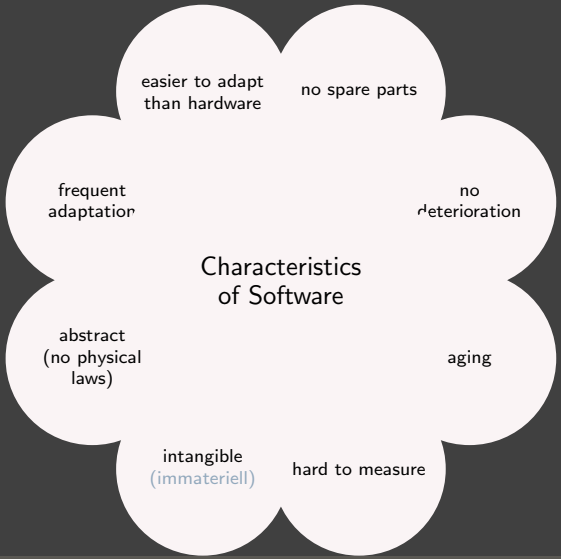
Lessons Learned

3. How Relevant is Software?

Why the Vasa Sank:



Characteristics of Software



Software (Products)

Software [adapted from Sommerville]

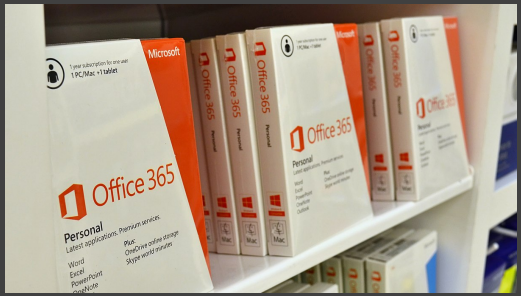
Software stands for one or several computer programs and all associated documentation, libraries, support websites, and configuration data that are needed to make these programs useful.

Explanation

The term program is used in a broader sense here. Software may also include source code, software models, or binaries.

Software Product and Professional Software

A **software product** is a software that can be sold to a customer. **Professional software** is software intended for use by someone apart from its developer and it is usually developed by teams rather than individuals. [adapted from Sommerville]



Application and System Software

Application Software or Application

Software that is designed for end users and applied for certain purposes. (Anwendungssoftware oder Anwendung)

Examples

web browsers, media players, email or chat clients, text or photo editors, games

System Software

Software that is not application software and typically being designed to provide a platform for other software.

Examples

operating systems, game engines, GUI frameworks

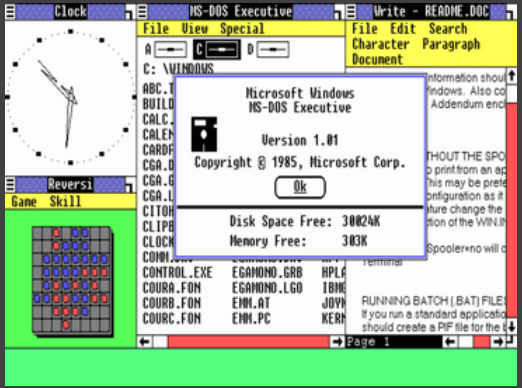
Classification Not Always Unique

e.g., web browsers and chat clients take over more and more features of operating systems

Application Software

Desktop Application or Desktop App

Windows 1.0 released in 1985

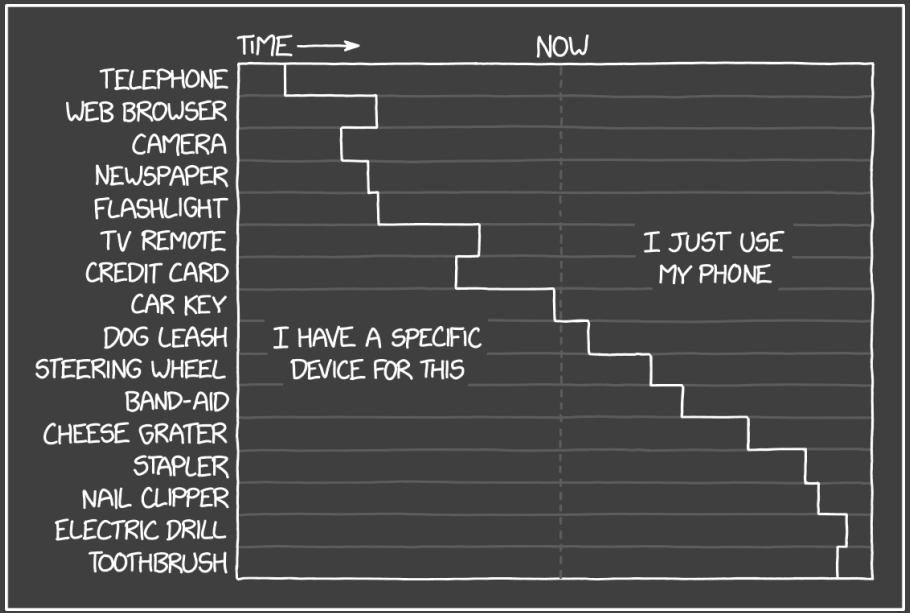


Web Application or Web App

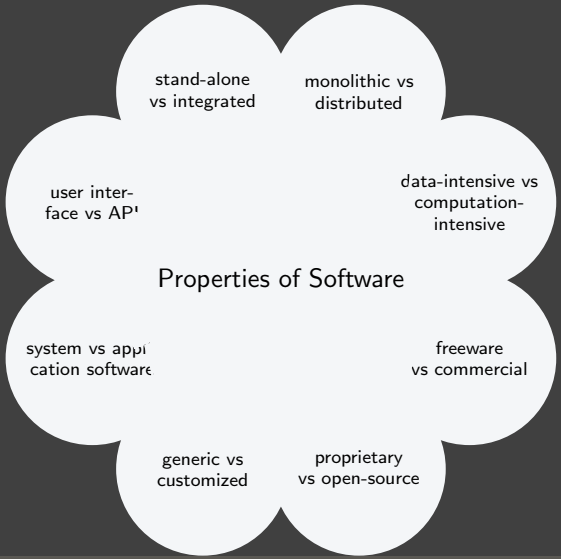
Ebay was born in 1995

Mobile Application or Mobile App or App

First iPhone released in 2007


TIME
NOW
TELEPHONE
WEB BROWSER
CAMERA
NEWSPAPER
FLASHLIGHT
TV REMOTE
CREDIT CARD
CAR KEY
DOG LEASH
STEERING WHEEL
BAND-AID
CHEESE GRATER
STAPLER
NAIL CLIPPER
ELECTRIC DRILL
TOOTHBRUSH
I JUST USE MY PHONE
I HAVE A SPECIFIC DEVICE FOR THIS

Properties of Software



Software Engineering

Software Engineering [Sommerville]

"Software engineering is an engineering discipline that is concerned with all aspects of software production from initial conception to operation and maintenance. [...] Software engineering is not just concerned with the technical processes of software development. It also includes activities such as software project management and the development of tools, methods, and theories to support software development."

Software Engineering vs Programming

what is needed besides programming will be motivated and shown throughout this course

HOW SOFTWARE DEVELOPMENT WORKS

Software Engineering vs Computer Science

SE vs CS [Sommerville]

"Computer science focuses on theory and fundamentals; software engineering is concerned with the practicalities of developing and delivering useful software. [...] Computer science theory, however, is often most applicable to relatively small programs. Elegant theories of computer science are rarely relevant to large, complex problems that require a software solution."

Software and System Engineering

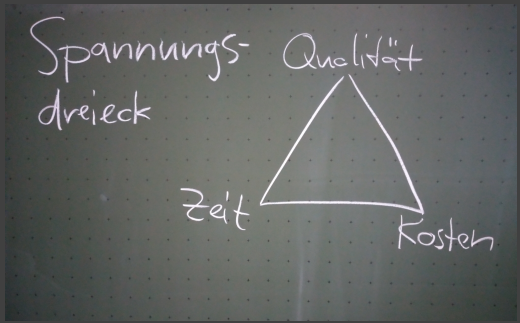
System Engineering [Sommerville]

"System engineering is concerned with all aspects of computer-based systems development including hardware, software and process engineering. Software engineering is part of this more general process."

(Software) Engineering

Engineering [Sommerville]

"Engineering is about getting results of the required **quality** within **schedule** and **budget**. [...] Engineers make things work. They apply theories, methods, and tools where these are appropriate. However, they use them selectively and always try to discover solutions to problems even when there are no applicable theories and methods. Engineers also recognize that they must work within organizational and financial constraints, and they must look for solutions within these constraints."



What is Software (Engineering)?

Lessons Learned

- What is software?
- What is the difference between program, software product, professional software, desktop/web/mobile app?
- What is software engineering?
- Which trade-off is crucial to software engineering?
- Further Reading: Sommerville, Chapter 1.1, p. 19–28

Practice

1. Form groups of 2-3 students
2. Introduce yourselves to each other (5 min)
3. Share your (recent) experiences with software (5 min)
4. Survey: What is your experience with software?

1. Introduction

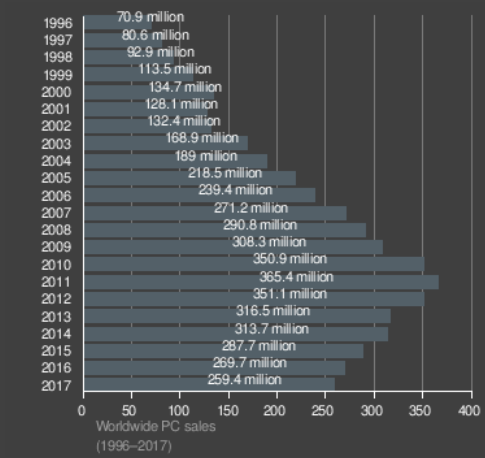
1. Course Introduction

2. What is Software (Engineering)?

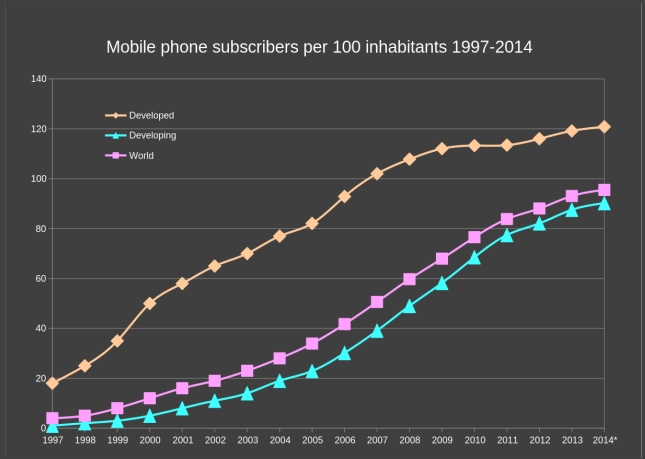
3. How Relevant is Software?

- World-Wide PC Sales
- World-Wide Mobile Phone Subscriptions
- Downloads of Android Apps
- Relevance of Software for This Course
- Lessons Learned

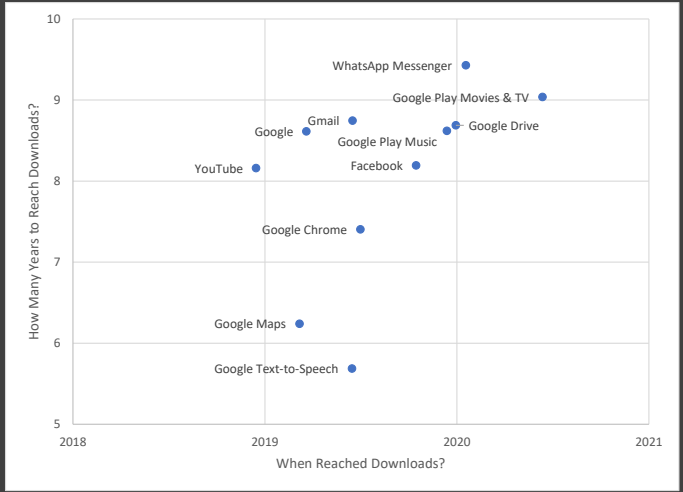
World-Wide PC Sales



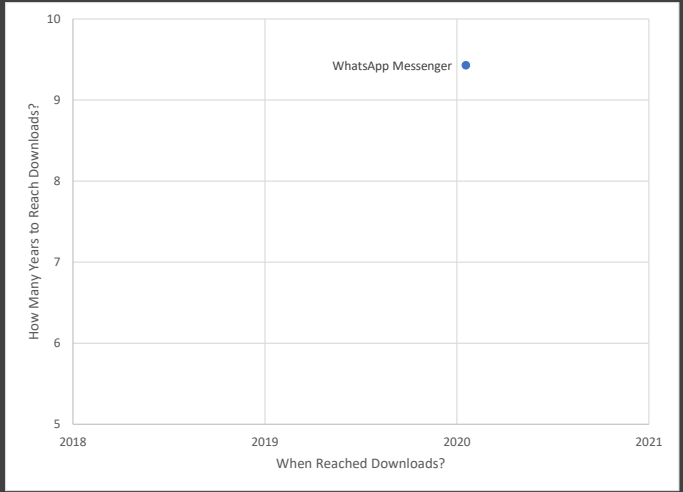
World-Wide Mobile Phone Subscriptions



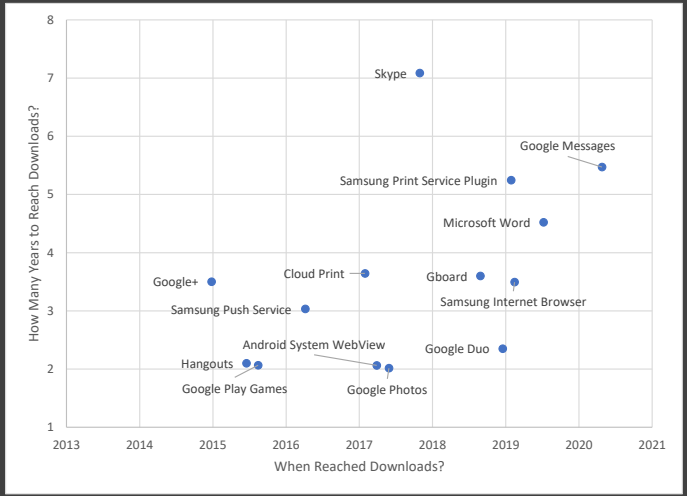
5 Billion Downloads of Android Apps



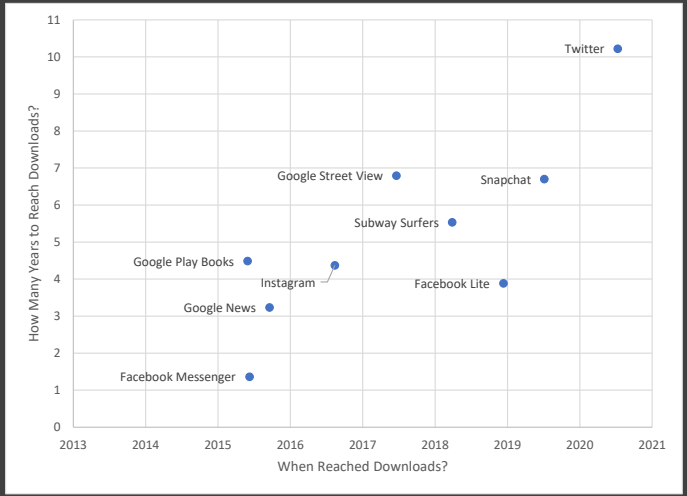
5 Billion Downloads of Android Apps



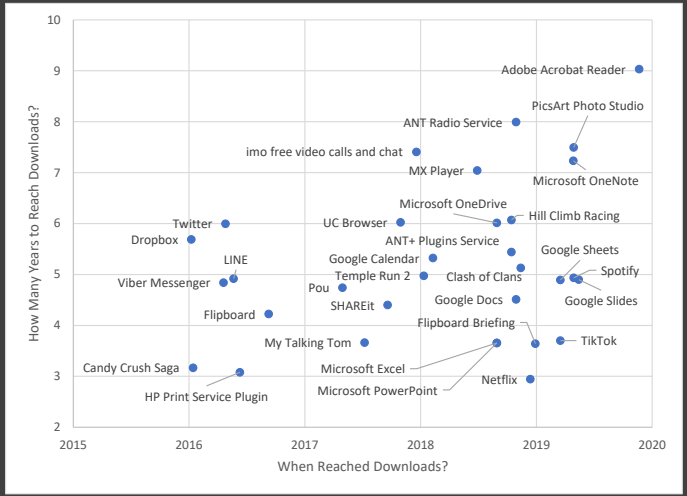
1 Billion Downloads of Android Apps



1 Billion Downloads of Android Apps



500 Million Downloads of Android Apps




ADOBE UPDATE
THERE IS AN UPDATE FOR:
• ADOBE DOWNLOAD MANAGER
THIS UPDATE WILL ALLOW YOU TO DOWNLOAD NEW UPDATES TO THE ADOBE UPDATE DOWNLOADER.
OK
DOWNLOAD

Relevance of Software for This Course

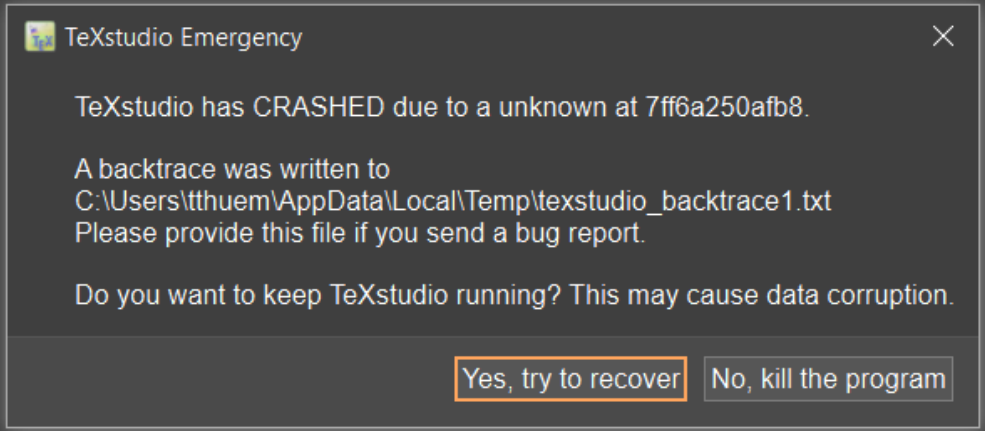
Google

500. That's an error.

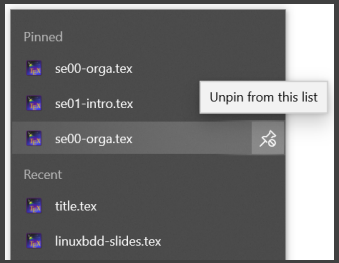
The server encountered an error and could not complete your request.

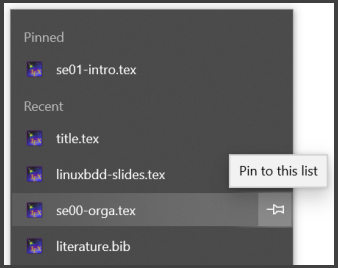
If the problem persists, please report your problem and mention this error message and the query that caused it.
That's all we know.

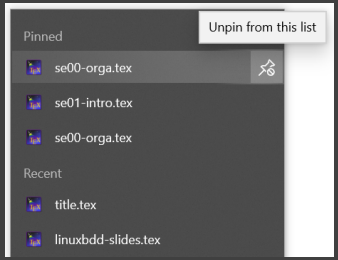
Relevance of Software for This Course



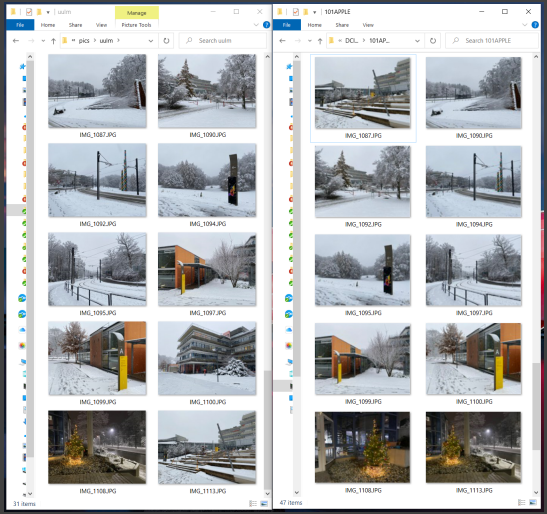
Relevance of Software for This Course



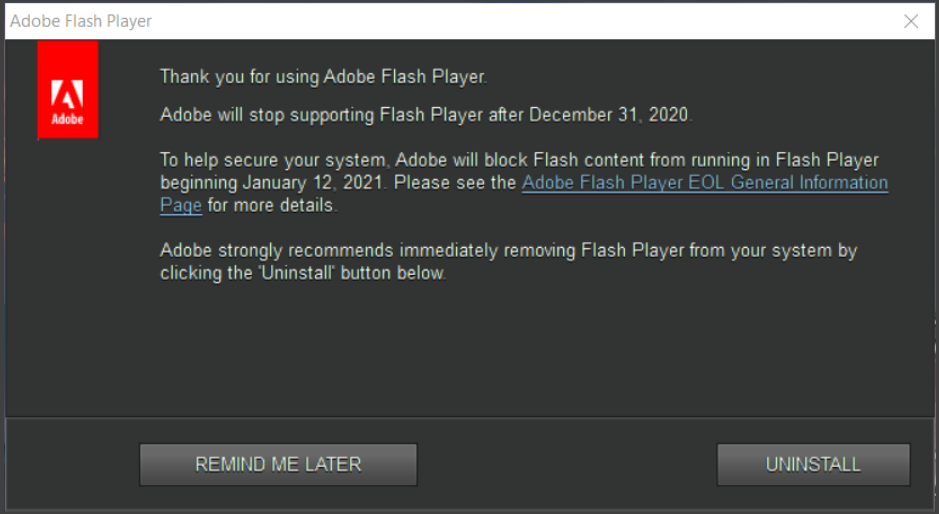




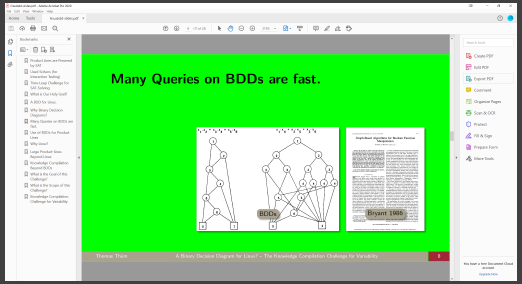
Relevance of Software for This Course

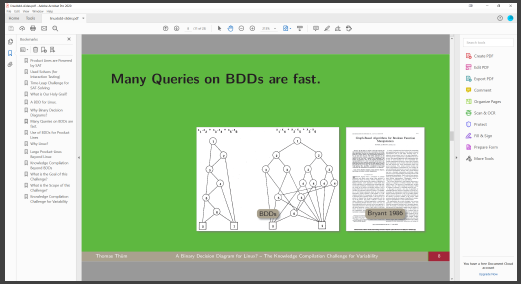


Relevance of Software for This Course

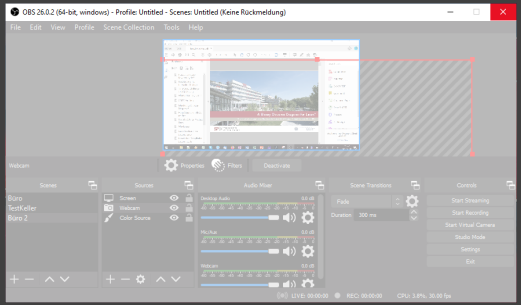


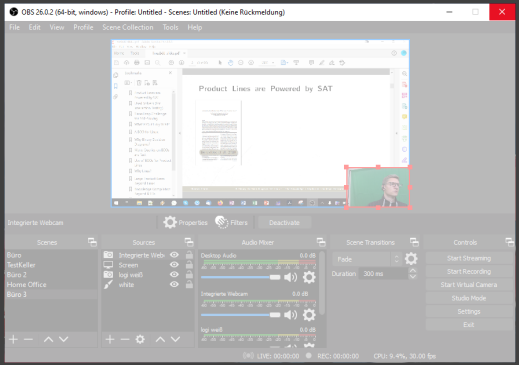
Relevance of Software for This Course



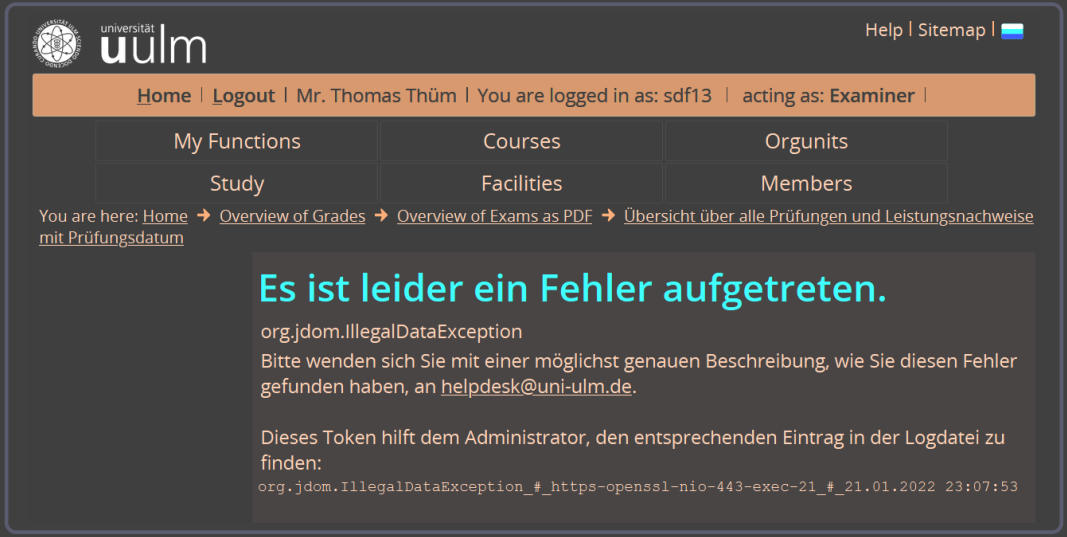


Relevance of Software for This Course

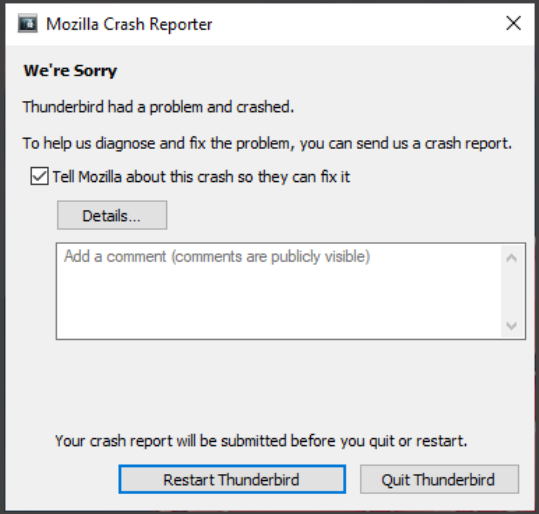




Relevance of Software for This Course



Relevance of Software for This Course



How Relevant is Software?

Lessons Learned

- What is the impact of software?
- How relevant is software for us?
- Next: How to develop the right thing?

Practice

1. Form groups of 2-3 students
2. Discuss: What will you do for a living in 10 years? What will be your connection to software? (5 min)
3. Survey: What is your connection to software in 10 years?