

Softwaretechnik / Software-Engineering

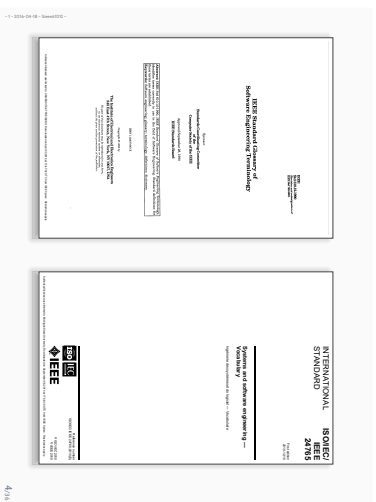
Lecture 1: Introduction

2006-04-18

Prof. Dr. Andreas Poddiski, Dr. Bernd Wespahl
Albert-Ludwigs-Universität Freiburg, Germany

Content

- Software Engineering, Software Engineering
- Successful Software Development
 - working definition: success
 - unsuccessful software development exists
 - common reasons for non-success
- Course
 - Content
 - topic areas
 - structure of topic areas
 - emphasis: formal methods
 - relation to other courses
 - literature
 - Organisation
 - lectures
 - tutorials
 - exam



4/18

Software – Computer program, procedures, and possibly associated documentation and data pertaining to the operation of a computer system.
See also: **application software**; **support software**; **system software**.
Context with: **hardware**.
IEEE 6190.1 (1993)

Software –
1. a collection of the programs, procedures, rules, and associated documentation of an information processing system. []
2. see 6190.12
3. program or set of programs used to run a computer. []
NOTE: Includes firmware, documentation data, and execution control statements.
IEEE 31945 (2010)

5/18

Software, Engineering, Software Engineering

Engineering vs. Non-Engineering

| | workshop (technical product) | studio (artwork, invention) |
|-------------------------------------|---|---|
| Motivation | the existing and available technical prerequisites | art's inspiration, art's idea |
| Deadlines | art's deadline is determined by the client with sufficient precision | art's deadline is determined by the artist's own will |
| Price | art's price is determined by the market value and by cost | art's price is determined by the artist's own will |
| Humor and Evaluation and comparison | art's humor and evaluation are determined by the artist's own will | art's humor and evaluation are determined by the artist's own will |
| Author | art's author is anonymous, art's author is not defined and is not clearly regulated with sufficient precision | art's author is anonymous, art's author is not defined and is not clearly regulated with sufficient precision |
| Warranty and liability | art's warranty and liability cannot be evaluated | art's warranty and liability cannot be evaluated |

(Ludwig and Lohr, 2013)

6/18

Software Engineering


Software Engineering –
(1) The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to software.
(2) The study of approaches as in (1).

Software Engineering –
1. the systematic application of scientific and technological knowledge, methods, and experience to the design, implementation, testing, and documentation of software.
2. see IEEE 610.12 (1)
ISO/IEC/IEEE 24765:2010

**Software Engineering:
Multi-person Development of Multi-version Programs.**
D. L. Parnas (2011)

Software Engineering – the establishment and use of sound engineering principles to obtain economically software that is reliable and works efficiently on real machines.

F. L. Bauer (1971)



7/36

The course's working definition of Software Engineering

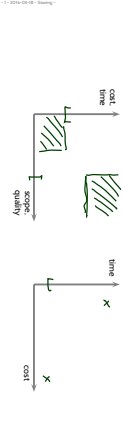
Software Engineering –

- (1) The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to software.
- (2) The study of approaches as in (1).

IEEE 6002 (1990)

Software Engineering – the establishment and use of sound engineering principles to obtain economically software that is reliable and works efficiently on real machines.

F. L. Bauer (1977)



9/36

Software Engineering—(1) The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to software.

(2) The study of approaches as in (1).

IEEE 630.1 (7/96)

Software Engineering – 1. the systematic application of scientific and technological knowledge, methods, and experience to the design, implementation, testing, and documentation of software.
2. see 6101216

ISO/IEC/JEKE 24765 (2.010)

Software Engineering Multi-person Development of Multi-version Programs.
D. L. Parnas (2011)

Software Engineering – the establishment and use of sound engineering principles to obtain economically software that is reliable and works efficiently on real machines. F. L. Bauer (1971)



853

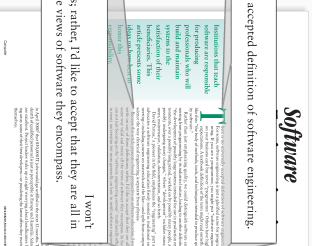
here is no univers

Software Engineering – 1, the systematic application of scientific and technological knowledge, methods, and experience to the design, implementation, testing, and documentation of software.

| |
|---------------------|
| Software version 01 |
|---------------------|

Software Engineering – the establishment and use of sound engineering principles to obtain economically software that is reliable and works efficiently on real machines.

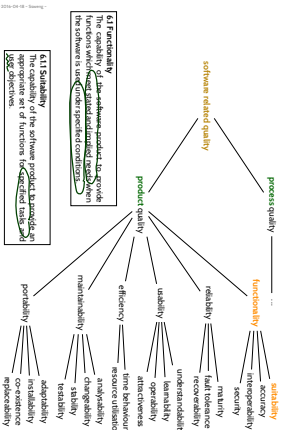
F. L. Bauer (1977)



8/1

"software that is reliable and works efficiently" (Bauer, 1971)

More general: software of (good) quality (cf. ISO/IEC 9126-1:2000 (2000))



10/3

gincering.

...and we could still spend our money on things that we like, but we would be spending it on things that we need. And we would be spending it on things that we need.

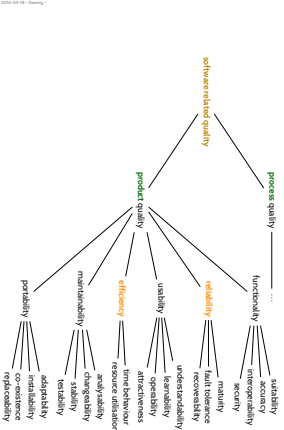
I won't

edges within in the same 12°

8/1

"software that is reliable and works efficiently" (Bauer, 1971)

More general software of (good) quality (cf. ISO/IEC 9126-1:2000 (2000)).



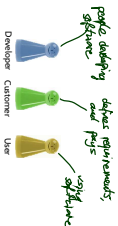
10.

Successful Software Development



11.1a

When is Software Development Successful?

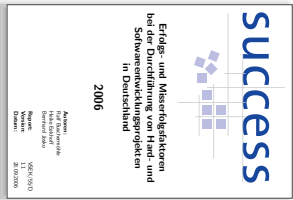


A software development project is **successful**
if and only if
developer, customer, and user are happy with the result at the end of the project



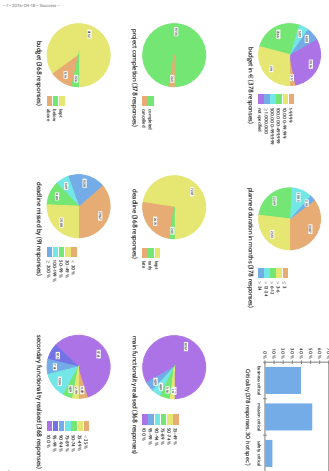
12.1a

Is Software Development Always Successful?



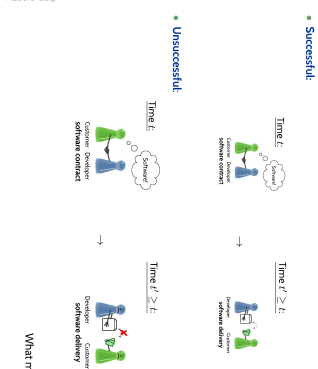
13.1a

Some Empirical Findings (Büchtemühle et al. (2006))

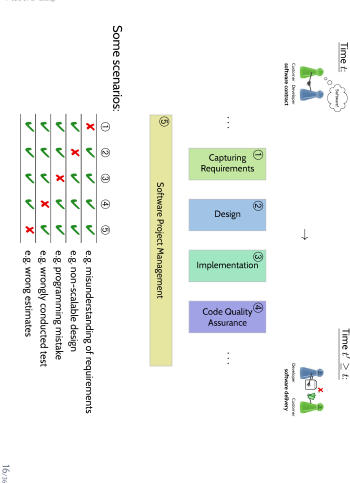


14.1a

A Closer Look



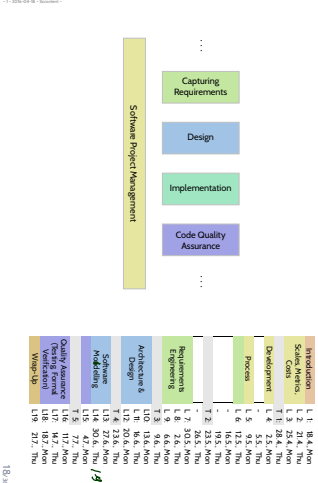
15.1a



16.1a

Course: Content

Course Content



Excursion: Informal vs. Formal Techniques

Example: Requirements Engineering, Abzug Controller

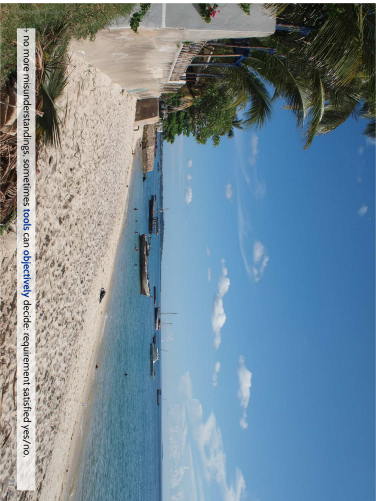


Requirement

Whenever a crash is detected, the abzug has to be fixed within 300 min. (F2).



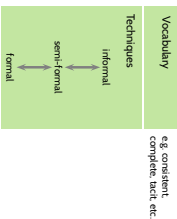
- vs.
- For observables: **crashdetected**: Time $\rightarrow \{0, 1\}$ and **fixabzug**: Time $\rightarrow \{0, 1\}$
 - Formalise requirement:
 $\forall t, t' \in \text{Time} \bullet \text{crashdetected}(t) \wedge \text{fixabzug}(t') \implies t' \in [t + 300 - \epsilon, t + 300 + \epsilon]$
- \rightarrow no more misunderstandings, sometimes tools can objectively decide requirement satisfied yes/no



no more misunderstandings, sometimes tools can objectively decide requirement satisfied yes/no

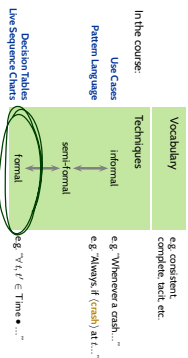
Structure of Topic Areas

Example: Requirements Engineering

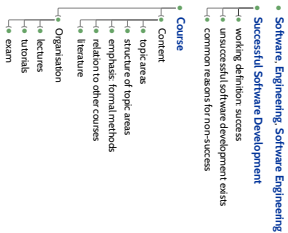


Structure of Topic Areas

Example: Requirements Engineering

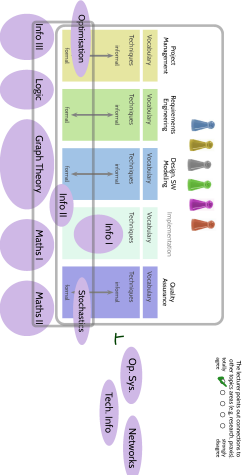


Content



23/16

Course Software-Engineering vs. Other Courses



24/16

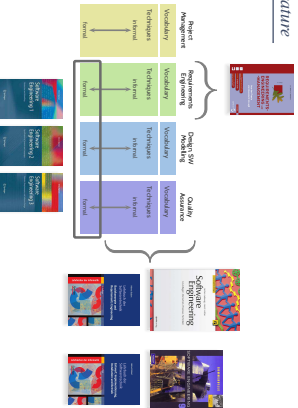
Course Software-Engineering vs. Softwarepraktikum

On popular demand:
The chair for software engineering is the strongest (on strongest) **practical** coupling between both courses.

| Chair | Software Engineering | Softwarepraktikum |
|-------|----------------------|-------------------|
| Chair | 1.1 18.4 Mio | 1.1 18.4 Mio |
| Chair | 1.2 21.4 Mio | 1.2 21.4 Mio |
| Chair | 1.3 21.4 Mio | 1.3 21.4 Mio |
| Chair | 1.4 21.4 Mio | 1.4 21.4 Mio |
| Chair | 1.5 21.4 Mio | 1.5 21.4 Mio |
| Chair | 1.6 21.4 Mio | 1.6 21.4 Mio |
| Chair | 1.7 21.4 Mio | 1.7 21.4 Mio |
| Chair | 1.8 21.4 Mio | 1.8 21.4 Mio |
| Chair | 1.9 21.4 Mio | 1.9 21.4 Mio |
| Chair | 1.10 21.4 Mio | 1.10 21.4 Mio |
| Chair | 1.11 21.4 Mio | 1.11 21.4 Mio |
| Chair | 1.12 21.4 Mio | 1.12 21.4 Mio |
| Chair | 1.13 21.4 Mio | 1.13 21.4 Mio |
| Chair | 1.14 21.4 Mio | 1.14 21.4 Mio |
| Chair | 1.15 21.4 Mio | 1.15 21.4 Mio |
| Chair | 1.16 21.4 Mio | 1.16 21.4 Mio |
| Chair | 1.17 21.4 Mio | 1.17 21.4 Mio |
| Chair | 1.18 21.4 Mio | 1.18 21.4 Mio |
| Chair | 1.19 21.4 Mio | 1.19 21.4 Mio |
| Chair | 1.20 21.4 Mio | 1.20 21.4 Mio |
| Chair | 1.21 21.4 Mio | 1.21 21.4 Mio |
| Chair | 1.22 21.4 Mio | 1.22 21.4 Mio |
| Chair | 1.23 21.4 Mio | 1.23 21.4 Mio |
| Chair | 1.24 21.4 Mio | 1.24 21.4 Mio |
| Chair | 1.25 21.4 Mio | 1.25 21.4 Mio |
| Chair | 1.26 21.4 Mio | 1.26 21.4 Mio |
| Chair | 1.27 21.4 Mio | 1.27 21.4 Mio |
| Chair | 1.28 21.4 Mio | 1.28 21.4 Mio |
| Chair | 1.29 21.4 Mio | 1.29 21.4 Mio |
| Chair | 1.30 21.4 Mio | 1.30 21.4 Mio |
| Chair | 1.31 21.4 Mio | 1.31 21.4 Mio |
| Chair | 1.32 21.4 Mio | 1.32 21.4 Mio |
| Chair | 1.33 21.4 Mio | 1.33 21.4 Mio |
| Chair | 1.34 21.4 Mio | 1.34 21.4 Mio |
| Chair | 1.35 21.4 Mio | 1.35 21.4 Mio |
| Chair | 1.36 21.4 Mio | 1.36 21.4 Mio |
| Chair | 1.37 21.4 Mio | 1.37 21.4 Mio |
| Chair | 1.38 21.4 Mio | 1.38 21.4 Mio |
| Chair | 1.39 21.4 Mio | 1.39 21.4 Mio |
| Chair | 1.40 21.4 Mio | 1.40 21.4 Mio |
| Chair | 1.41 21.4 Mio | 1.41 21.4 Mio |
| Chair | 1.42 21.4 Mio | 1.42 21.4 Mio |
| Chair | 1.43 21.4 Mio | 1.43 21.4 Mio |
| Chair | 1.44 21.4 Mio | 1.44 21.4 Mio |
| Chair | 1.45 21.4 Mio | 1.45 21.4 Mio |
| Chair | 1.46 21.4 Mio | 1.46 21.4 Mio |
| Chair | 1.47 21.4 Mio | 1.47 21.4 Mio |
| Chair | 1.48 21.4 Mio | 1.48 21.4 Mio |
| Chair | 1.49 21.4 Mio | 1.49 21.4 Mio |
| Chair | 1.50 21.4 Mio | 1.50 21.4 Mio |
| Chair | 1.51 21.4 Mio | 1.51 21.4 Mio |
| Chair | 1.52 21.4 Mio | 1.52 21.4 Mio |
| Chair | 1.53 21.4 Mio | 1.53 21.4 Mio |
| Chair | 1.54 21.4 Mio | 1.54 21.4 Mio |
| Chair | 1.55 21.4 Mio | 1.55 21.4 Mio |
| Chair | 1.56 21.4 Mio | 1.56 21.4 Mio |
| Chair | 1.57 21.4 Mio | 1.57 21.4 Mio |
| Chair | 1.58 21.4 Mio | 1.58 21.4 Mio |
| Chair | 1.59 21.4 Mio | 1.59 21.4 Mio |
| Chair | 1.60 21.4 Mio | 1.60 21.4 Mio |
| Chair | 1.61 21.4 Mio | 1.61 21.4 Mio |
| Chair | 1.62 21.4 Mio | 1.62 21.4 Mio |
| Chair | 1.63 21.4 Mio | 1.63 21.4 Mio |
| Chair | 1.64 21.4 Mio | 1.64 21.4 Mio |
| Chair | 1.65 21.4 Mio | 1.65 21.4 Mio |
| Chair | 1.66 21.4 Mio | 1.66 21.4 Mio |
| Chair | 1.67 21.4 Mio | 1.67 21.4 Mio |
| Chair | 1.68 21.4 Mio | 1.68 21.4 Mio |
| Chair | 1.69 21.4 Mio | 1.69 21.4 Mio |
| Chair | 1.70 21.4 Mio | 1.70 21.4 Mio |
| Chair | 1.71 21.4 Mio | 1.71 21.4 Mio |
| Chair | 1.72 21.4 Mio | 1.72 21.4 Mio |
| Chair | 1.73 21.4 Mio | 1.73 21.4 Mio |
| Chair | 1.74 21.4 Mio | 1.74 21.4 Mio |
| Chair | 1.75 21.4 Mio | 1.75 21.4 Mio |
| Chair | 1.76 21.4 Mio | 1.76 21.4 Mio |
| Chair | 1.77 21.4 Mio | 1.77 21.4 Mio |
| Chair | 1.78 21.4 Mio | 1.78 21.4 Mio |
| Chair | 1.79 21.4 Mio | 1.79 21.4 Mio |
| Chair | 1.80 21.4 Mio | 1.80 21.4 Mio |
| Chair | 1.81 21.4 Mio | 1.81 21.4 Mio |
| Chair | 1.82 21.4 Mio | 1.82 21.4 Mio |
| Chair | 1.83 21.4 Mio | 1.83 21.4 Mio |
| Chair | 1.84 21.4 Mio | 1.84 21.4 Mio |
| Chair | 1.85 21.4 Mio | 1.85 21.4 Mio |
| Chair | 1.86 21.4 Mio | 1.86 21.4 Mio |
| Chair | 1.87 21.4 Mio | 1.87 21.4 Mio |
| Chair | 1.88 21.4 Mio | 1.88 21.4 Mio |
| Chair | 1.89 21.4 Mio | 1.89 21.4 Mio |
| Chair | 1.90 21.4 Mio | 1.90 21.4 Mio |
| Chair | 1.91 21.4 Mio | 1.91 21.4 Mio |
| Chair | 1.92 21.4 Mio | 1.92 21.4 Mio |
| Chair | 1.93 21.4 Mio | 1.93 21.4 Mio |
| Chair | 1.94 21.4 Mio | 1.94 21.4 Mio |
| Chair | 1.95 21.4 Mio | 1.95 21.4 Mio |
| Chair | 1.96 21.4 Mio | 1.96 21.4 Mio |
| Chair | 1.97 21.4 Mio | 1.97 21.4 Mio |
| Chair | 1.98 21.4 Mio | 1.98 21.4 Mio |
| Chair | 1.99 21.4 Mio | 1.99 21.4 Mio |
| Chair | 2.00 21.4 Mio | 2.00 21.4 Mio |

25/16

Literature



... more on the course homepage.

26/16

Any Questions So Far?

Course: Organisation

28/16

References

Bauer, F. L. (1971). Software engineering. In *IFIP Congress (I)*, pages 530–538.

Buchterreible, R., Eickhoff, H. and Jöcks, B. (2004). *success – Erfolgs- und Misserfolgsfaktoren bei der Durchführung von Hard- und Softwareentwicklungsprojekten in Deutschland*. Technical Report VSEI/55/D.

IEEE (1990). *IEEE Standard Glossary of Software Engineering Terminology*. Std 610.12-1990.

ISO/IEC FDIS (2000). *Information technology - Software product quality - Part 1: Quality model*. 9126-1:2000(E).

ISO/IEC (IEEE) (2010). *Systems and software engineering - Vocabulary*. 24765:2010(E).

Ludewig, J. and Lichte, H. (2013). *Software Engineering*. dpunkt-verlag, 3. edition.

Parnas, D. L. (2011). Software engineering: Multi-person development of multi-version programs. In *Parnas, C. B. et al., editors, Dependable and Historic Computing*, volume 6875 of *LNCS*, pages 413–427. Springer.