# Software Engineering

Summer 2012

Everything on these slides can also be found on the Web site: http://www.st.cs.uni-saarland.de/edu/se/2012/

## A Software Crisis

**Denver International Airport (DIA)**

Construction started in 1989 • 53 sq miles • Planned: 1.7 bio USD costs, opening 1993

## Denver International Airport

- Approved for construction in 1989
- First major airport to be built in the United States in over 20 years.
- Three terminals + several runways
- Built on 53 square miles of land (Twice the size of Manhattan Island!)

# BAE Contract

- Original assumption: Every company builds its own baggage transport system
- United (70% Denver traffic) was the only to begin planning; contract with BAE
- First fully automated baggage system
- Later, Denver airport extended contract to entire airport – three times original size

# The Scope

- 20 miles of track
- 6 miles of conveyor belts
- 56 laser arrays that read bar coded tags
- 400 frequency readers
- 3,100 standard size baggage 'Telecars'
- 450 6.5 ft by 4 ft oversize cars
- 55 separate computers

# The System

# The Timeframe

- BAE started work 17 months before scheduled opening October 31, 2003
- In Munich (similar system), engineers had spent *two years* just *testing* the system (with 24/7 operation six months before the airport opened)

# More Risks

- Most of buildings were already done, so BAE had to accommodate system (sharp turns, narrow corridors…)
- BAE paid little attention to German sister project and devised system from scratch
- Little communication within BAE

# Final Blunder

- The decision to broadcast the preliminary test of the "revolutionary" new baggage system on national television

While all of the previous blunders contributed to the failure of the system, the worst mistake was still yet to come. It was a decision so terrible, so foolish, so absolutely

What camera crews depicted was truly a disaster; carts jammed together, damaged luggage everywhere, some bags literally split in half, and the tattered remains of clothing strewn

## A Disaster

- Carts jammed together
- Damaged luggage everywhere, some bags literally split in half
- Tattered remains of clothing strewn about caused subsequent carts to derail
- Half the luggage that survived the ordeal ended up at the wrong terminal

## More Issues

- Carts got stuck in narrow corridors
- Wind blew light baggage from carts
- 5% of the labels were read correctly
- Normal network load was 95%

# Complexity: Empty Carts

- Empty carts need to go where they are needed
- Cart has to be at its "cannon" at the right moment
- Lanes have limited length ➔ traffic jam
- All controlled by single central system

# Consequences

- Airport opening delayed four times – overall, sixteen months late
- New engineering firm
  - split system in three (one per terminal)
  - implemented manual backup system
- BAE got bankrupt
- Overall damage: 1.3 bln USD

# Glass' Law

*Requirement deficiencies* are the prime source of project failures.

This and other laws are found in Endres/Rombach: Handbook of Software and Systems Engineering. Evidence: Denver airport case study and two more

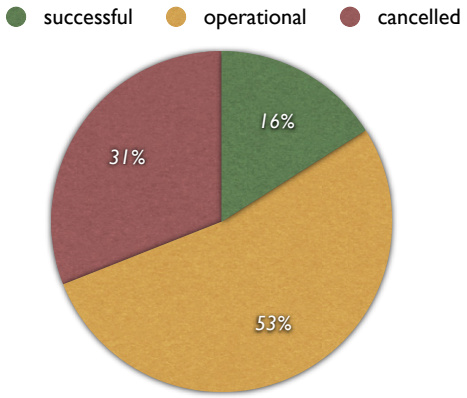
# Chaos Report

- 31% of projects were *aborted* prior to completion
- in small (large) development companies, *only 16% (9%) of all projects* were completed within projected budget and time limits

Survey by the Standish Group, 1994 – 350 companies with >8000 software projects

http://www.standishgroup.com

# Project Success



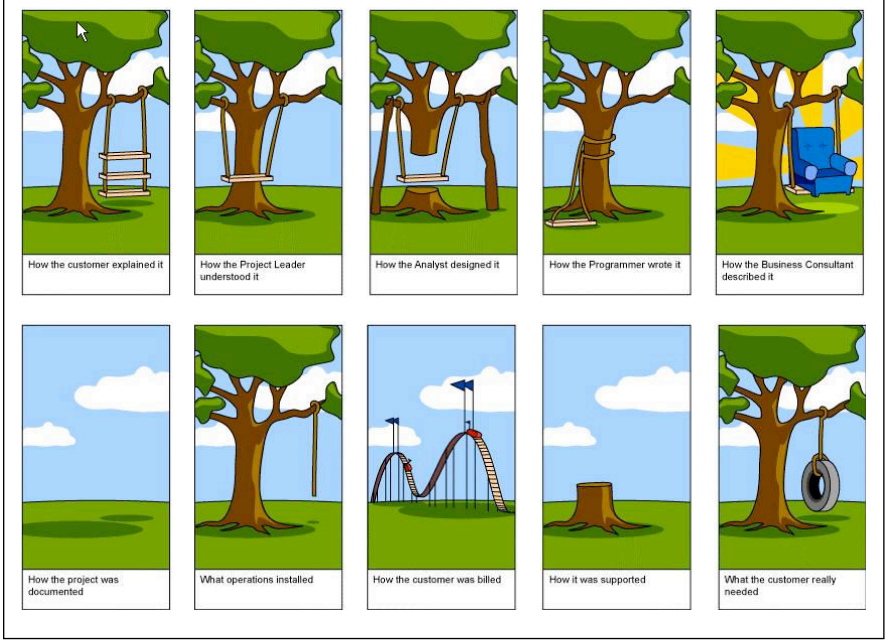

Survey by PC week, 1995: 365 information systems professionals on success of software development projects

16% of all projects successful
53% operational, but less than successful
31% *cancelled*

# More Examples

- Mariner 1 (1962)
  Rocket crash due to missing dash
- Eole 1 (1971)
  72 weather balloons get wrong cmd
- Nimbus 7 (1978)
  Satellite misses ozone hole for 6 yrs
- HMS Sheffield (1982)
  Exocet rocket id'ed as "friend"
- Stanislaw Petrow (1983)
  Russia detects global nuclear attack
- Therac 25 (1985)
  Radiation overdose kills six
- Stock crash (1987)
  Dow Jones loses 22% in one day
- Vincennes (1988)
  Passenger jet mistaken to be F-14
- Patriot (1991)
  Misses to shoot down Iraqi Scud
- Climate Orbiter (1999)
  Confuses metrics and imperial
- US Blackout (2003)
  50 mln affected for 5 days
- Social support (2004)
  No money for millions

http://www.tagesanzeiger.ch/digital/computer/13-Softwarefehler-die-zu-Katastrophen-fuehrten/story/21703807




# Challenges

- Why does it take so long to get software finished?
- Why are the development costs so high?
- Why can't we find all errors?
- Why do we spend so much time and effort maintaining existing programs?
- Why is it difficult to measure progress?

# Topics

- Requirements Engineering
- Software Specification
- Software Design and Architecture
- Software Quality Assurance
- Software Maintenance and Evolution
- Software Project Management

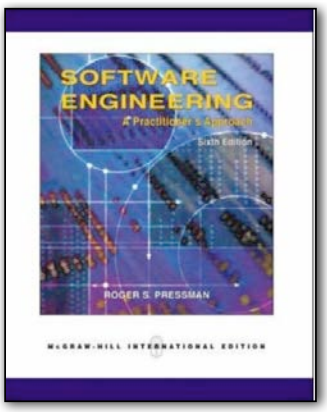
# Your Lecturers

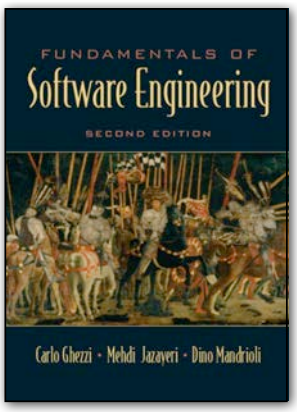
- Andreas Zeller
- Lecture Tue/Thu 08:30–10:00 HS 002
- Irregular timing (see Web page)

# Your Tutors

- Florian Gross (coordinator)
- Konrad Jamrozik (senior tutor)
- Aliaksandr Talaika
- Max-Ferdinand Suffel
- Daria Gaidar
- Mykola Havrikov
- Frederik Leonhardt
- Olga Mykytiuk
- Krishna Narasimhan
- Sabina Glowacka
- Marie-Therese Walter
- Souza N. Windiartono

# Books



# Exam


(+ extra exam mid September)

# Projects

- SW Engineering is best learned by *doing*
  (There is no "theory of software engineering")
- Therefore, *projects* make up 2/3 of course

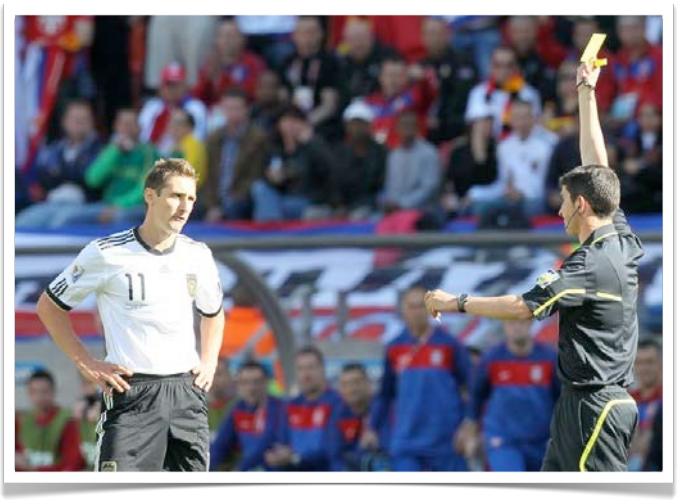
# Projects

# Client

# Team

# Tutor

# Supervision



# Honor

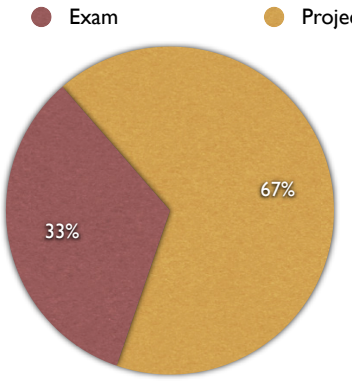
# Project Details

- Non-trivial piece of software
- Suggested by *client* (a CS member)
- Client is *busy* (spends max 15 hrs total)
- Client is *vague* (on purpose)

# Deliverables

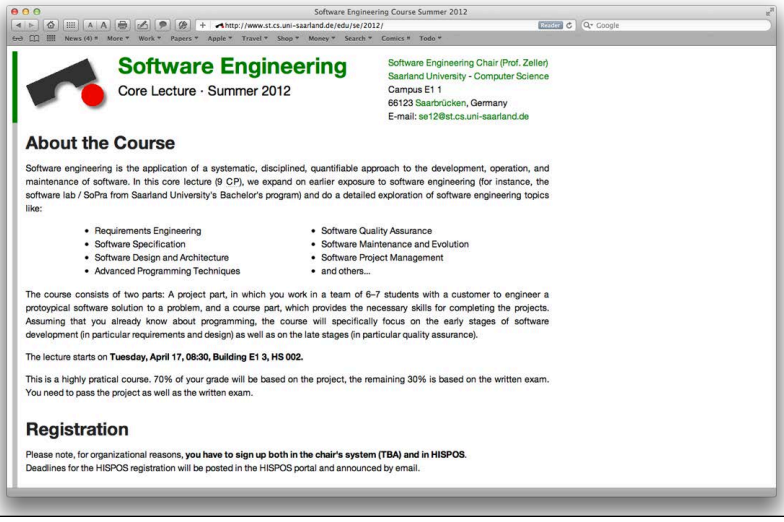
- Full set of *requirements*
- User interface design
- Architecture design
- Project plan
- Prototype

# Grading




- Need to pass *both* exam and project to pass
- Project grades based on group performance (with bonus for individuals)

# Web Site




http://www.st.cs.uni-saarland.de/edu/se/2012/

# Sign up!

irements Engineering
ware Specification
ware Design and Architecture
nced Programming Techniques

- Software Quality Assurance
- Software Maintenance and Evolution
- Software Project Management
- and others...

of two parts: A project part, in which you work in a team of 6–7 students with a customer t
solution to a problem, and a course part, which provides the necessary skills for completing
already know about programming, the course will specifically focus on the early stages
ular requirements and design) as well as on the late stages (in particular quality assurance).

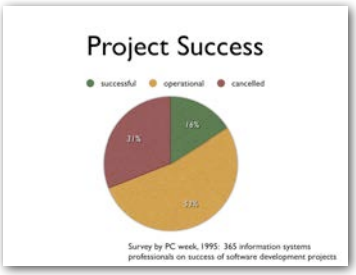
uesday, April 17, 08:30, Building E1 3, HS 002.

l course. 70% of your grade will be based on the project, the remaining 30% is based on the
project as well as the written exam.

on

zational reasons, **you have to sign up both in the chair's system (TBA) and in HISPOS.**
OS registration will be posted in the HISPOS portal and announced by email.


# Summary




## Challenges

- Why does it take so long to get software finished?
- Why are the development costs so high?
- Why can't we find all errors?
- Why do we spend so much time and effort maintaining existing programs?
- Why is it difficult to measure progress?