

Software Engineering

- Application of systematic, disciplined, quantifiable approach to software development, operation and maintenance of software. (*IEEE-Std.*)
- The term *software engineering* was used occasionally in the late 1950s and early 1960s. It was popularized as a response to the software crisis during the 1968 NATO Software Engineering Conference (held in Garmisch, Germany) by its chairman F.L. Bauer, and has been in widespread use since.

Why do we need SE?

- Increased demands for software with:
 - high quality
 - high complexity
 - delivered promptly
 - low price
- Frequent failures:
 - Space shuttle
 - Nike
 - Denver airport

Kr51

Charette, R.N. (2005) Why Software Fails, IEEE Spectrum, September 2005

Challenges to SE

- | | |
|--|---|
| <ul style="list-style-type: none"> ■ Different types of software: <ul style="list-style-type: none"> ■ criticality ■ # users ■ # developers ■ # platforms ■ risk ■ cost ■ There will be changes <ul style="list-style-type: none"> ■ degrading performance ■ maintenance account for 70% of life cycle cost ■ Everyone want the latest technology | <ul style="list-style-type: none"> ■ Different types of knowledge: <ul style="list-style-type: none"> ■ Understanding customers ■ Understanding users ■ Managing projects and people ■ Communication ■ Staffing ■ Design at many levels ■ Programming tools ■ Components ■ Testing ■ Quality assurance ■ Business ■ Risk management ■ Technology assessment ■ |
|--|---|

Kr52

My view

Make processes, but also components, tools, people,... fit nicely together



Always ask what you need and what you can get.
The rest is like finishing a jigsaw-puzzle.
(The Boeing 777-200 has about 1400 data processing units and 5 million lines of code.)

This course will present you some basic SE-elements mostly suited for 10¹-10² people projects.

Slide 2

KrS1 What is your favourite software failure story?
Kristian Sandahl; 2006-01-23

Slide 3

KrS2 There is also a research crisis.
Kristian Sandahl; 2006-01-23