

Introduktion til Software Engineering

Michael R. Hansen

mrh@imm.dtu.dk

Informatics and Mathematical Modelling
Technical University of Denmark

02161 Software Engineering 1 ©Michael R. Hansen, Spring 2008 – p.1/9

Introduktion

Software Engineering

- den disciplin, der vedrører alle aspekter ved produktion af programmel, herunder specifikation, konstruktion, test, udvikling, drift og vedligeholdelse.

Begrebet 'software engineering' blev indført i forbindelse med en konference i 1968.

Emne 'the software crisis'

Mål programmeringskunsten skulle blive til en ingeniørdisciplin

Det er, trods fremskridt, stadig udfordrende at lave programmel,

- der opfylder brugernes forventninger, og
- er afleveret til tiden og indenfor budgettet.

ung og umoden disciplin

02161 Software Engineering 1 ©Michael R. Hansen, Spring 2008 – p.2/9

Nogle definitioner

- F.L. Bauer, from *Information Processing*, 1972.
"The establishment and use of sound engineering principles (methods) in order to obtain economically software that is reliable and works on real machines"
- IEEE (Institute of Electrical and Electronics Engineers)
"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)
"

02161 Software Engineering 1 ©Michael R. Hansen, Spring 2008 – p.3/9

Nogle forhold om programudvikling

- store programmer
- korrekte programmer
- hyppige modifikation af programmer
- konstrueret af flere, muligvis store, udviklings teams.
- ...

Desværre er det ofte sådan at

- tidsplaner skrider (meget)
- omkostningerne skrider (meget)
- grænser flyttes
- vanskeligt at producere software af god kvalitet
- ...

02161 Software Engineering 1 ©Michael R. Hansen, Spring 2008 – p.4/9

Hvorfor er SE svært?

- Krav ændres og tilføjes hyppigt
- Vanskeligt at indfange de rigtige krav
- Svært at overskue andres og egne programmer
- Svært at gøre rede for programmers korrekthed
- Begrænsninger i metoder til test af programmel
- Begrænsninger i metoder til vedligeholdelse af programmel
- Vanskeligt at lave programdokumentation af høj kvalitet
- Vanskeligt at vedligeholde programdokumentation
-
-
-

Hvad kan man gøre?

02161 Software Engineering 1 ©Michael R. Hansen, Spring 2008 – p.5/9

Typiske faser i et projekt

- **Analyse** af krav Kravspecifikation
— samarbejde med brugere
- Modellering og **design** Designspecifikation
— begrebsmodellering, komponentopdeling, grænseflader
- **Implementering** Teknisk dokumentation
— kode, komponenttest, algoritmevalg, ...
- **Test** Test skema for systemet

Mange andre typiske faser findes, f.eks. **vedligeholdelse**, men kun de fire ovennævnte vil optræde i 02161.

Der findes en række paradigmer for den process det er at udvikle software. I disse **udviklingsprocesser** kombineres faserne på forskellige måder.

Udfordring: **rød tråd (sporbarhed)** fra krav til kode

02161 Software Engineering 1 ©Michael R. Hansen, Spring 2008 – p.6/9

Kursusmål

- At kunne gennemføre et “mindre projekt” ud fra en “uformel og åben beskrivelse” til et produkt
- At kunne udvikle krav, design, implementering og test således at væsentlige begreber kan spores igennem hele forløbet

02161 Software Engineering 1 ©Michael R. Hansen, Spring 2008 – p.7/9

Udfordringer i 02161

- at **skabe begreber** ud fra en uformel beskrivelse af en anvendelse
- at udvikle begreberne igennem de fire **abstraktionsniveauer** kravspecifikation, design, implementering og kode
- at redegøre for specificationer (krav, design, ...) og program

02161 Software Engineering 1 ©Michael R. Hansen, Spring 2008 – p.8/9

Kursus indhold

- opstille en kravspecifikation i form af use case beskrivelser på baggrund af en given problemstilling.
- anvende UMLs klasse- og sekvensdiagrammer til beskrivelse af programdesign
- design-by-contract
- systematiske teknikker til planlægning af test
- programmeringsteknikker
- brug af udviklingsværktøjer
- gennemføre et mindre software projekt