

Introduksjon til systemutvikling og innhold i forelesningsplanen

Universitetslektor Yngve Lindsjørn

Institutt for Informatikk, UiO

Yngve Lindsjørn (yngve.lindsjorn@hioa.no)

- Universitetslektor ved Ifi, UiO
- Mastergrad, Ifi, UiO, 1987
- Forsker ved Norsk Regnesentral (IT-forskning) 1987-97
- Drev eget programvareselskap, Kompetanseweb, i 8 år
- Lang erfaring fra næringslivet – med blant annet prosjektledelse for store IKT-prosjekter
- Forsker på effektivitet i smidige team

Oppgaveundervisning, lab og obligatoriske oppgaver

- **Kristin Brænden-** kristinbraenden@gmail.com
 - Studerer Informatikk ved Ifi
- **Emilie Hallgren** - emiliehallgren@gmail.com
 - Studerer Informatikk ved Ifi

Temaer i dagens forelesning

- Hva systemutvikling (software engineering) handler om
- Hvilke temaer dere skal lære
- Undervisningsopplegg

Hva er systemutvikling?

- Systemutvikling (software engineering) starter når noen interessenter (“stakeholders”) ønsker et datasystem for å håndtere utfordringer eller utnytte potensialet i en virksomhet
- Systemutvikling handler om hvordan man utvikler, installerer, forbedrer, tilpasser og utvider IT-systemer av tilstrekkelig **kvalitet** innen akseptable **tids-** og **kostnadsrammer**
- Faget systemutvikling omfatter teorier, metoder, modeller og verktøy for utvikling av datasystemer

Hvorfor lære systemutvikling?

- Programmering er sentralt, men andre aktiviteter er også nødvendige i systemutvikling, f.eks.
 - Analyse av problemområdet (kunde og utvikler må bli enige om hva som skal lages og til hvilken kvalitet)
 - Design av systemet
 - Testing (validering og verifisering) av systemet
 - Installering hos kunder eller bruker
- Måten man jobber på, dvs. (utviklings)prosessen er viktig

Datasytemer

- Programvaresystemer = software-systemer = datasytemer = IT-systemer
- Slike systemer danner bærebjelken i de økonomiske, politiske, sosiale, kulturelle og vitenskapelige sfærene av det moderne informasjonssamfunnet
- Noen av dem er noe av det mest komplekse som menneskeheten har produsert



Masseprodukter vs. skreddersydde systemer

- Masseprodukter
 - Selvstendige, markedsførte "hyllevare"-systemer": tekstbehandlingsprogrammer, prosjektstyringsverktøy, spill etc.
 - Firmaene som utvikler systemene, bestemmer hva som skal lages og hvilken endringer som skal implementeres
- Skreddersydde eller spesialtilpassede systemer
 - Systemer som spesifiseres og bestilles av en bestemt kunde eller oppdragsgiver: skreddersydde systemer for bedrifter eller offentlige systemer for skatt, flykontroll, miljøovervåkning etc.
 - Kunden lager og eier spesifikasjonen for systemet og bestemmer endringer som skal implementeres

Typer av systemer

- **Uavhengige systemer:** Kjører lokalt på en datamaskin, f.eks. PC, og trenger ikke koples i noe nettverk
- **Interaktive transaksjonssystemer:** Grensesnittet mot brukerne er på en PC eller terminal, men systemer kjøres på datamaskiner som står andre steder. Eks. Web-applikasjoner
- **Innebygde (embedded) systemer:** programvare som kontrollerer og styrer maskinvare. Eks. mobiltelefoner, togkontrollsystemer
- **Batch-systemer:** Forretningskritiske systemer som prosesserer data i store bolker. Eks. bank og finans
- **Underholdningssystemer:** Spill og annen underholdning
- **Systemer for modellering og simulering:** utviklet av forskere og ingeniører for å modellere fysiske prosesser eller situasjoner
- **Data-innsamlingssystemer:** samler inn data fra omgivelsene ved bruk av sensorer og sender dataene til andre systemer for behandling. Eks. værstasjoner, satellitter

Porteføljer: systemer av systemer

- Er ofte “ultra-large scale systems”, dvs.
 - ekstremt komplekse
 - endres kontinuerlig
 - mange interessegrupper
 - mange ulike maskinvareprodukter, etc.
- Eksempler:
 - World Wide Web
 - internasjonal flytrafikk-kontroll
 - porteføljen til store, multinasjonale selskaper, f eks. har Telenor flere hundre IT-systemer i sin portefølje



Begrepene *safety* og *security*

- Generelt på norsk
 - *Safety*: sikkerhet mot uønskede hendelser som resultat av tilfeldigheter
 - *Security*: sikkerhet mot uønskede hendelser som resultat av overlegg
- Innen systemutvikling
 - Safety: Det skal ikke være risikabelt å bruke datasystemer (sikkerhet)
 - Security: Datasystemer skal hindre at de selv eller deres data blir angrepet utenfra (sikring)

Hvilke konsekvenser har det at systemer er av ulike typer?

- Ulike typer systemer har ulike egenskaper og stiller ulike typer krav
- Systemene må derfor utvikles på ulike måter
- Hva vil være forskjellig ved utvikling av et fly/tog-kontrollsystem og web-system som gir oversikt over HiOA's kurs?



Variasjon i størrelse

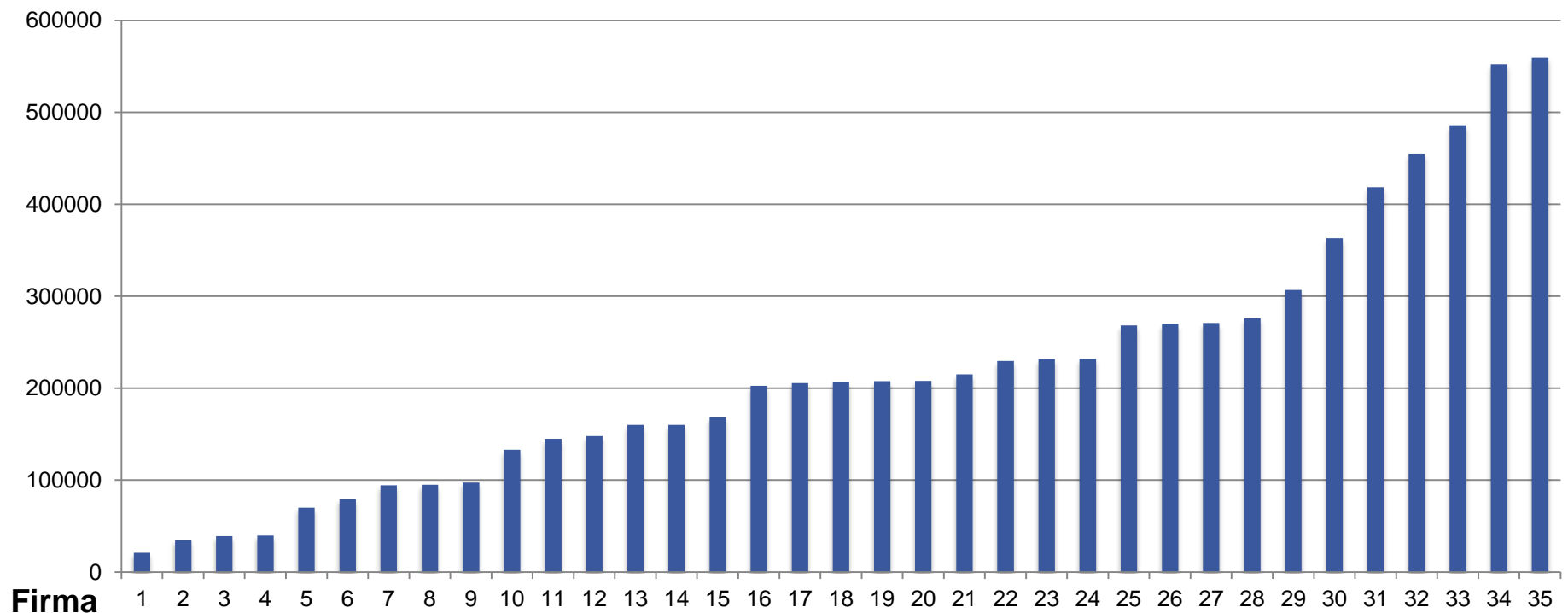
- Programvaren
 - Fra noen få tusen til mange millioner linjer kode
- Utviklingsteamene
 - fra enkeltpersoner til over 1000 utviklere (MS Windows 7)
- Kostnad utvikling og vedlikehold
 - fra noen tusen kroner til flere milliarder

Variasjon i systemutvikling i Norge

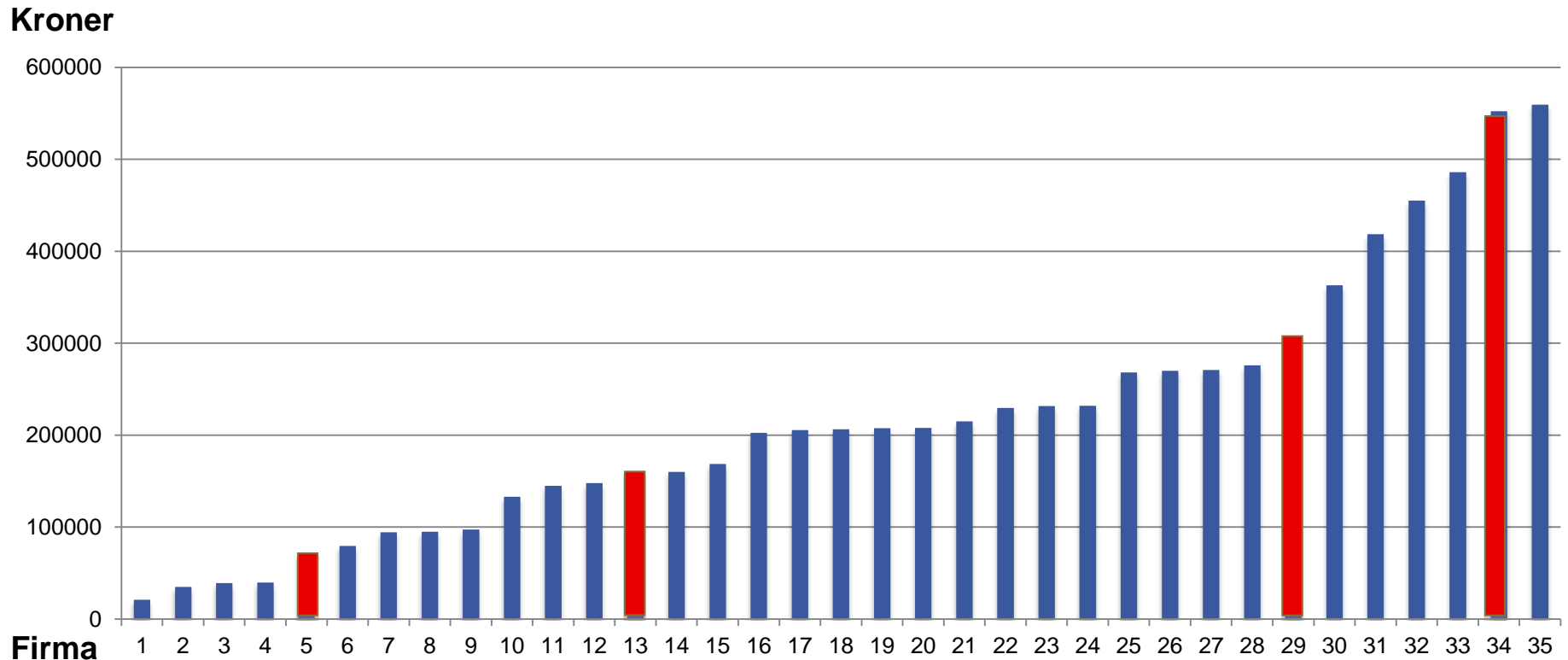
- 81 firmaer invitert til å komme med anbud på et lite web-basert informasjonssystem
- 35 firmaer la inn anbud
- Hvor stor variasjon i pristilbud?

Fra kr. 21.000 til kr. 560.000 !

Kroner

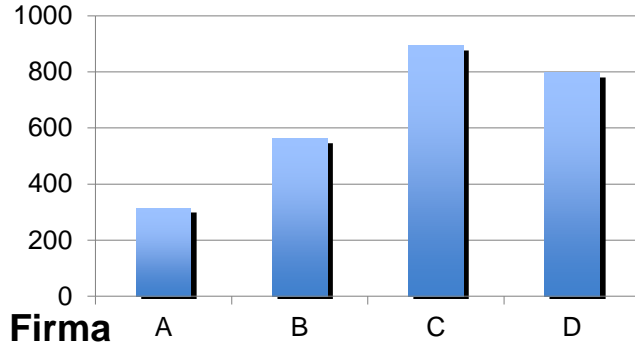


Hvilket firma ville dere ha valgt?

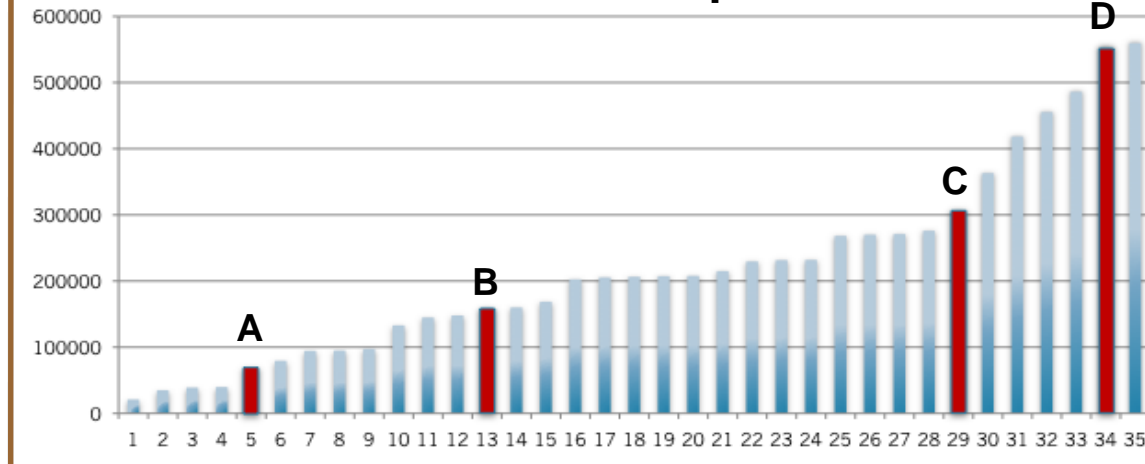


Resultat

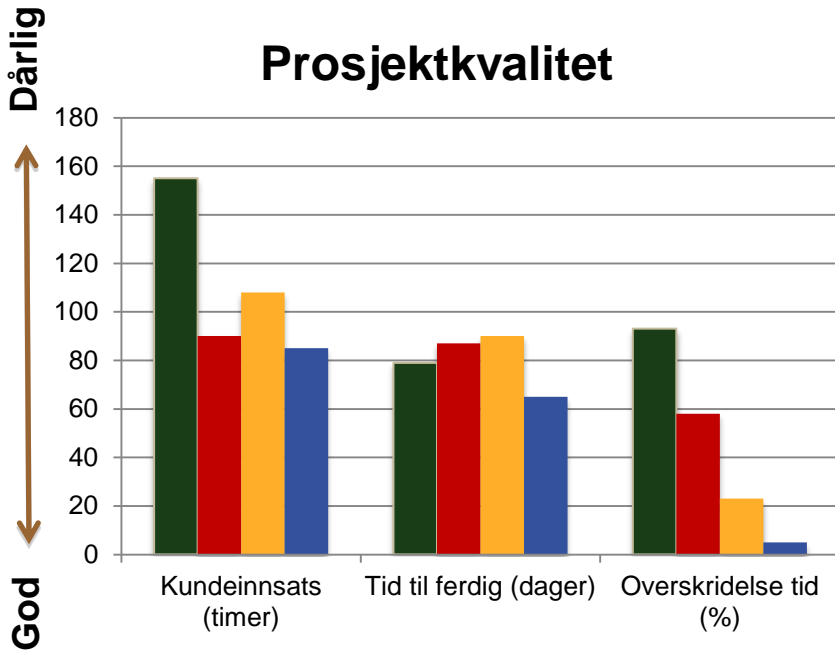
Timer



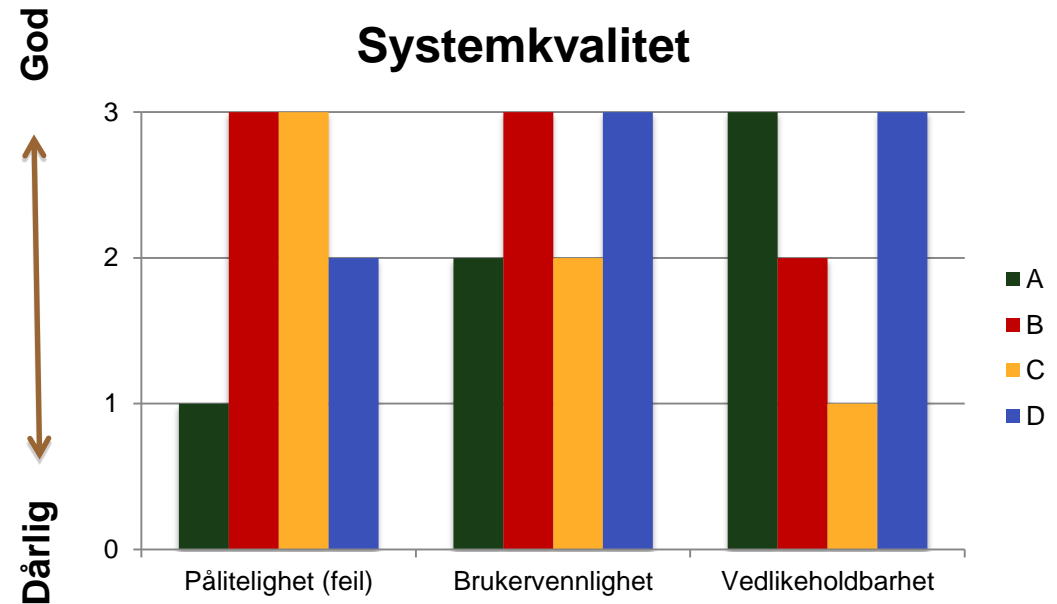
Anbudspris



Prosjektkvalitet



Systemkvalitet



Hva er årsakene til forskjellene?

- Ulike måter å jobbe gir ulike resultater
- Din kompetanse og måten du og ditt team jobber på vil avgjøre hvordan prosjektet og sluttproduktet blir!
- Derfor er temaene i dette systemutviklingskurset viktige

Systemutvikling (software engineering)

– en ingeniørdisiplin

- Bidrar til at vi lager bedre systemer med færre ressurser på en raskere og mer forutsigbar måte
- Betrakter både menneskelige og teknologiske aspekter
- Baseres på ingeniørprinsipper (evidensbaserte/empiriske systematiske metoder) med fokus på:
 - Planlegging og forutsigbarhet (i motsetning til “ta den tiden som trengs”)
 - Oppdeling og strukturering av problemer i mindre komplekse deler (i motsetning til ”prøv og feil”)
 - Modularitet og gjenbruk (i motsetning til “lag alt fra bunnen av hver gang”)
 - Abstraksjon og modellering (i motsetning til ”bare koden er systemet”)
 - Systematisk kvalitetssikring (i motsetning til “gjør som du selv synes er best”)

5 hovedprinsipper

Planlegging og forutsigbarhet

- Veldefinerte, repeterbare og planlagte aktiviteter
 - Alle personer vet hva de skal gjøre, hvordan det gjøres (standarder/metoder/verktøy), hva og når de skal levere
- Prosjektplaner og -rapportering
 - Ressursplaner: Kostnadsrammer, personal, utstyr
 - Tidsplaner: Estimering, milepæler, aktivitetsnettverk
- Kvalitetsplaner og -rapportering
 - Sjekklistor, inspeksjoner, testplaner, testresultater ...
 - Rutiner for å håndtere endringsforespørsler, sporbarhet, ...
- Men graden av planlegging og formalitet i systemutvikling er et kontroversielt tema
 - Planbaserte versus smidige metoder

Oppdeling og strukturering av problemer i mindre komplekse deler

- Prosessaktiviteter
 - Analyse, design, programmering (koding), testing, installering, vedlikehold/videreutvikling, ...
- Egenskaper ved systemet
 - Funksjonalitet, kvalitetsaspekter
- Modeller på forskjellige abstraksjonsnivåer
 - Kravspesifikasjoner versus objekt-orienterte designmodeller versus kode

Modularisering og gjenbruk

- Modularisering innebærer at datasystemer deles opp i mindre delsystemer (komponenter, moduler)
 - Hvert delsystem implementerer et veldefinert problem (høy kohesjon)
 - Reduserer avhengigheter på tvers av delsystemer (lav kobling)
 - Muliggjør gjenbruk innen et prosjekt eller på tvers av prosjekter
 - Lettere arbeidsfordeling og samarbeid
 - Forenkler inkrementell og smidig utvikling
 - Komponentbasert utvikling
 - Finnes det standardprogrammer eller standardkomponenter som allerede dekker (deler av) behovene eller som enkelt kan tilpasses til å dekke behovene?

Abstraksjon og modellering

- Identifiser de viktigste momentene og ignorer irrelevante detaljer
- Modeller er abstraksjoner: spesifikasjoner og designmodeller skjuler irrelevante programmeringsdetaljer
 - Selve programmet kan også ses på som en presis modell av hvilke oppgaver som skal gjøres og hvordan, som deretter oversettes til maskinkode slik at datamaskinen kan utføre dem
- Dere vil lære om Unified Modeling Language (UML)
 - UML brukes til å spesifisere kravene til et system og analyse hvordan disse kravene kan realiseres i et objekt-orientert programmeringsspråk

Systematisk kvalitetssikring

- Kundeinvolvering
- Validering og verifisering
 - Validering: Har vi spesifisert det riktige systemet, dvs. er det dette kunden ønsker eller trenger?
 - Verifisering: Er systemet riktig, dvs. har vi minimert antall feil?
- Smidig utvikling
 - Reduserer risiko ved at man leverer og evaluerer (validerer og verifiserer) delsystemer fortløpende, tett kundekontakt
- Endringshåndtering og konfigurasjonsstyring

Profesjonalitet

- Respekter intellektuelle rettigheter (IPR) som copyrights og patenter til arbeidsgivere og klienter
- Ikke misbruk andres maskiner (spekter fra spill til skadelig virus)
- Respekter konfidensiell informasjon du måtte få hos arbeidsgivere eller klienter uansett om du har signert en konfidensialitetserklæring eller ikke
- Forhold deg til din egen kompetanse, dvs. ikke påta deg arbeid som er klart utenfor ditt kompetanseområde

Etikk innen systemutvikling

- Ansvar utover bare anvendelsen av teknisk kunnskap og om å følge loven – handle moralsk riktig
- Ved å utvikle datasystemer kan utviklere bidra til goder eller skade, enten ved å utvikle selv eller påvirke andre
- For å bidra til gode formål må utviklere ta ansvar for å gjøre systemutvikling til en positiv og respektert profesjon

Etiske dilemmaer

- Du mener arbeidsgiver har innført metoder og formelle prosesser for systemutvikling som ikke gir gode systemer
- Arbeidsgiver opptrer uetisk ved å installere et sikkerhetskritisk system hos en kunde før testingen er ferdig
- Delta i utvikling av militære våpensystemer eller produkter som er klima- eller miljø-ødeleggende

Kompetanseområder innen systemutvikling

Domenekunnskap

Virksomhetsforståelse

Utviklingsprosesser

Kravhåndtering

Prosjektstyring

Avtaler og kontrakter

Estimering

Jus og etikk

Arkitektur

Kvalitetssikring/Testing

Design

Endringshåndtering

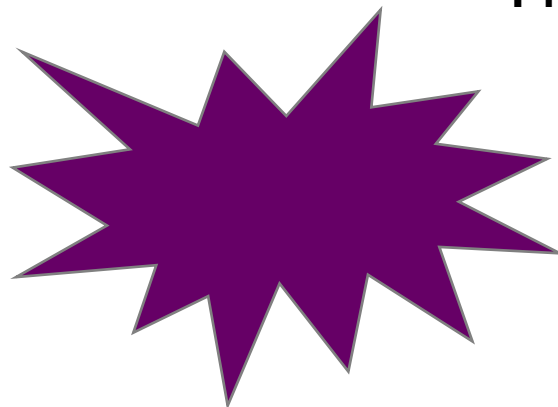
Modellering

Konfigurasjonsstyring

Programmering

Databaser

Utviklingsverktøy



Evaluering av når de ulike metodene og teknologiene fungerer

Næringsliv, forvaltning og industri etterspør kompetanse i alle disse feltene!

Kursets faglige elementer og forelesere

Dato	Emne	Foreleser	Pensum
18.08	Innledning, hva systemutvikling er, oversikt over kurset.	Yngve Lindsjørn Rom:PH330	Lærebok kap 1 IS kap. 1
25.08	Systemutviklingsprosessen, prosessmodeller. Smidig programvareutvikling.	Yngve Lindsjørn	Lærebok kap 2 og 3.1-3.5 IS kap. 2 og 3
01.09	Kravhåndtering pluss use-case modellering (UML)	Yngve Lindsjørn	Lærebok kap. 4 og 5 IS kap. 4 og 5
08.09	Fra krav til objekter (UML)	Yngve Lindsjørn	Lærebok kap. 5, 6 IS kap. 5, 7
15.09	Objektorientert design ved bruk av UML. Klassediagrammer. Interface. Design Patterns	Yngve Lindsjørn	Lærebok kap. 5, 6 IS kap. 5, 7
22.09	Arkitektur	Suhas Joshi	Lærebok kap. 7 IS kap. 6
29.09	Testing	Suhas Joshi	Lærebok kap. 8 IS kap. 8
06.10	Prosjektledelse, teamarbeid og prosjektplanlegging	Yngve Lindsjørn	Lærebok. kap 9, 10 IS kap . 22, 23
13.10	Estimering	Johas Sushi	Lærebok kap. 10 IS kap. 23
27.10	Repetisjon – med vekt på UML	Yngve Lindsjørn	
03.11	Metoder for evaluering av systemutvikling. Forskningsmetoder	Yngve Lindsjørn	Lærebok kap 3.6, 4.5.2, 4.5.5 IS kap. 4.5.5.
10.11	Kontrakter	Ståle Hagen	
17.11	Gjennomgang eksamensoppgave	Yngve Lindsjørn	
24.11	Avklarende spørsmål, tips til eksamen.	Yngve Lindsjørn	

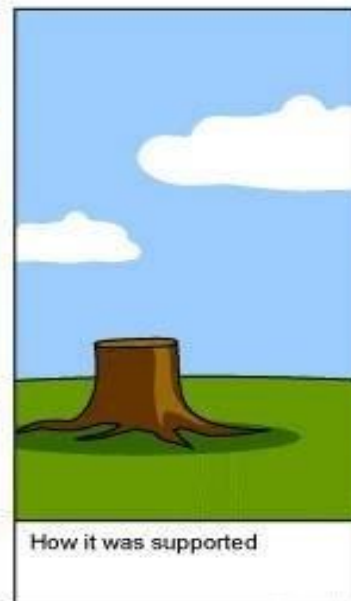
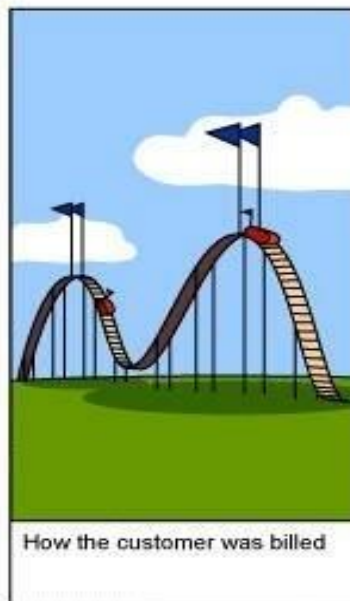
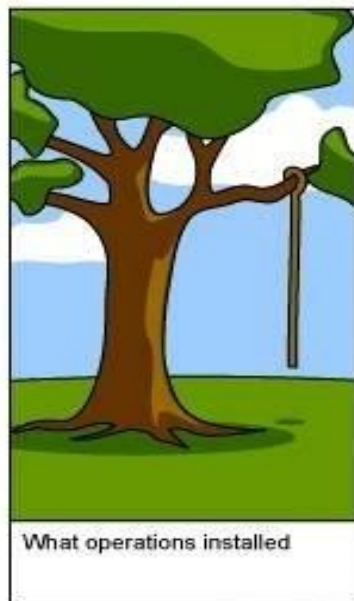
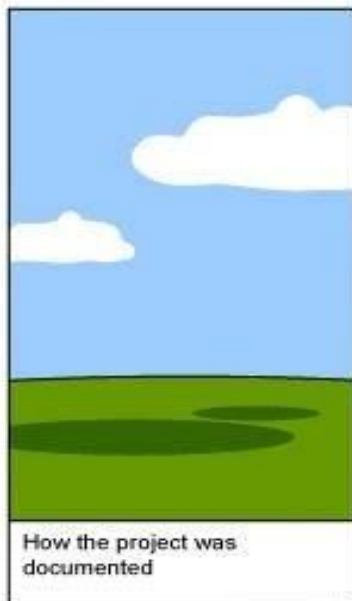
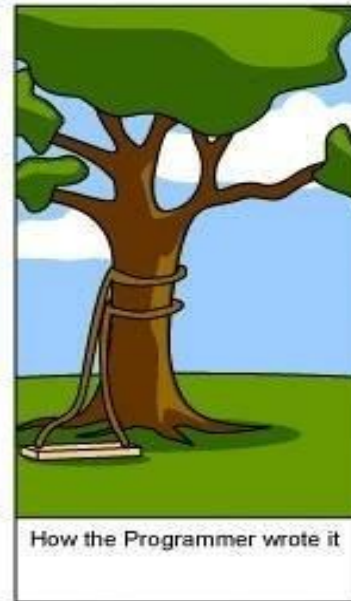
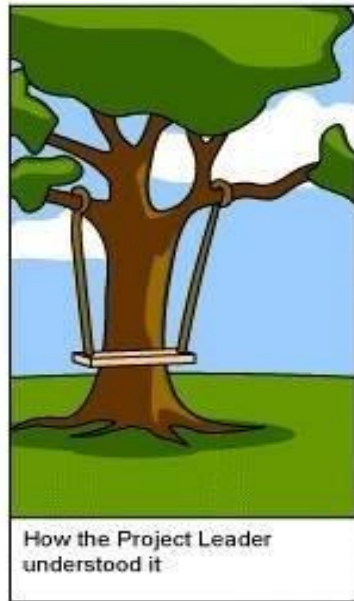
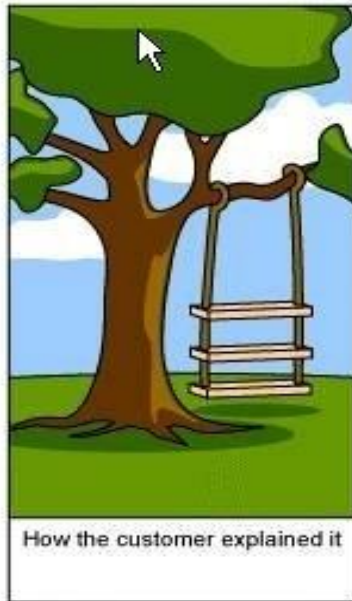
Systemutviklingsprosessen

- Utviklingsprosessen (= måten man jobber på) i et utviklingsprosjekt påvirker kvaliteten både på prosjektet og systemet som utvikles
 - Prosjektet
 - Tid, kostnader, brukerinvolvering
 - Produktet (systemet)
 - Funksjonalitet (hva gjør systemet), nytte, pålitelighet, ytelse, vedlikeholdbarhet etc.
- Måten man jobber på påvirker også arbeidsmiljøet (trivsel, motivasjon, kompetanseutvikling etc.) som igjen påvirker prosjekt- og produktkvalitet
- Prosessmodeller: fossefall, inkrementell, smidige metoder etc.

Smidige metoder og prosjektledelse

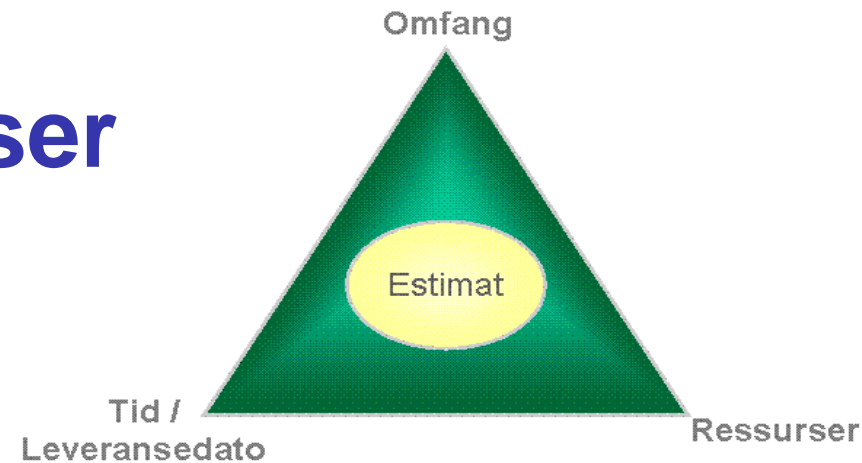
- I praksis klarer man ikke spesifisere alle kravene til et system av en viss størrelse i forkant
- Derfor er såkalt smidig utvikling blitt utbredt: Grunnleggende prinsipper og illustrert ved konkrete metoder og teknikker
 - Plandrevet vs smidig utvikling
 - Teamarbeid
 - Scrum
 - Kanban
- Ledelse og styring av systemutviklingsprosjekter, kontrakter

Kravhåndtering



Estimering av systemutviklingsprosesser

- Man bør kunne si noe om kostnader og tid ved utvikling av et IT-system
 - Når kan kunde/brukere forvente å ta systemet i bruk?
 - Hvor mye penger må skaffes for å utvikle systemet?
 - Hva slags funksjonalitet og andre krav ønsker man å nedprioritere hvis tid og penger ikke strekker til?
- Metoder og teknikker for estimering
 - “tradisjonelle” modeller
 - Metoder ofte brukt i smidig utvikling (planning poker, relativ-estimering, story points etc.)



Modellering – bruk av UML

- System-modellering
 - hvordan grafiske modeller kan representere programvaresystemer
- Unified Modelling Language (UML)
 - Use Case, sekvensdiagram, klassediagram, aktivitetsdiagram og tilstandsdiagram

Arkitektur

- Programvarearkitektur og arkitekturdesign
- Hvorfor arkitekturdesign er viktig i programvareutvikling

Testing

- Utviklingstesting
 - Enhetstesting, systemtesting
- Testdrevet utvikling
 - Spesielt innen smidig utvikling
- “Release”-testing
- Brukertest

Pensum

- Lærebok
 - Ian Sommerville: Software Engineering. 9 ed. (Pearson Education Ltd. /Addison-Wesley), kap. 1-8, 22 og 23, (deler av 19)
 - Selected chapters from Software Engineering. Hele boka
- Lysark fra forelesningene

Oppgaver og obligatorikse oppgaver

- Ukeoppgaver – felles gjennomgang fredager
 - Følger stort sett tema fra mandagens forelesning
 - Se på oppgavene før gjennomgang
 - Får hjelp på lab på torsdager
- Tre obligatoriske oppgaver
 - Deles inn i grupper
 - Vil få hjelp på lab

Gjennomføring

- Forelesningene vil fokusere både på det vi mener er mest sentralt og på stoff som utfyller læreboka
- Forelesningene gir mulighet til diskusjoner og spørsmål
- Studer undervisningsplan, les kapitlene i læreboka, samt forelesningslysark (legges ut hver uke) i Fronter
- Ukeoppgaver legges ut etter forelesningen mandag – senest onsdag

Vurderingsform

- Tre godkjente obligatoriske oppgaver
 - Krav for å gå opp til eksamen
- Skriftlig eksamen (3 timer)
 - Ingen hjelpemidler tillatt
 - Bestemmer karakteren

Du har også et ansvar for undervisningen!

- Vis omtanke! Forelesere og medstudenter forventer at:
 - du er på plass i det forelesningen begynner
 - du ikke småprater med andre under forelesningen
 - du ikke spiller spill, surfer, chatter, e-mailer o.l under forelesningen
 - du gjør alt du kan for å engasjere deg og følge med i forelesningen
 - selv om det iblant kan være tungt!

Kontakt

- Faglige spørsmål og kommentarer: (går til kursansvarlig: Yngve Lindsjørn)
- Spørsmål om oppgaver, obligatoriske oppgaver, lab-arbeid etc. går til Emilie Hallgren og Kristin Brænden
- Studieadministrative spørsmål: (går til studieadministrasjonen – 7 etasje)