

# Kravhåndtering

Erik Arisholm  
Simula Research Laboratory  
&  
Institutt for informatikk

© Institutt for informatikk – Erik Arisholm (med innspill fra Ray Welland) 04.02.2009 — INF1050-krav-1

# Kravhåndtering

- ❑ **Kravhåndtering** (innsamling, analyse og en mer eller mindre presis spesifikasjon av kravene til et system) er en sentral del i de aller fleste utviklingsprosjekter, uavhengig av hvilken utviklingsprosess som brukes
  - Engelsk: Requirements Engineering
- ❑ **Kostnadene** ved å rette feil i kravene etter systemleveranse er svært høy
  - kanskje 100 ganger mer enn kostnadene ved å rette en kodefeil
  - *Standish*-rapporten: Mangelfull kravhåndtering var rapportert som viktigste årsak (37 %) til at utviklingsprosjekter fikk problemer
- ❑ **Endringer i krav** er uungåelig
  - etter hvert som vår forståelse av kravene bedres
  - etter hvert som utviklingen skrider fram
  - etter at systemet er levert (vedlikehold, evolusjon)



© Institutt for informatikk – Erik Arisholm (med innspill fra Ray Welland) 04.02.2009 — INF1050-krav-2

# Kravenes rolle i systemutviklingsprosessen

- ❑ De fleste kravspesikasjonsmetoder antar at hoveddelen av kravene bør identifiseres, spesifiseres og valideres (godkjennes) så godt som mulig før selve utviklingen (design, koding, testing) starter.
- ❑ Lettvektsprosesser (agile methods) forsøker å integrere kravhåndteringen i selve utviklingsløpet.



© Scott Adams, Inc./Dist. by UFS, Inc.

© Institutt for informatikk – Erik Arisholm (med innspill fra Ray Welland) 04.02.2009 — INF1050-krav-3

# Noen definisjoner

- ❑ **Interessant** - en eller annen som har en direkte eller indirekte interesse av at systemet utvikles, og som derfor påvirker kravene til et system.
  - Engelsk: Stakeholder
  - Interessenter kan være oppdragsgiver, kunder, lovgivere, brukergrupper, systemeiere...
- ❑ **Funksjonelle krav** - beskriver oppførselen (funksjonene) til systemet
  - Engelsk: System services, functional requirements
- ❑ **Ikke-funksjonelle krav** - legger føringer på oppførselen i form av krav til ytelse, sikkerhet, brukervennlighet, kostnader, tidsrammer, interoperabilitet, osv
  - Engelsk: System constraints, non-functional requirements



© Institutt for informatikk – Erik Arisholm (med innspill fra Ray Welland) 04.02.2009 — INF1050-krav-4

## Kravhåndtering er en systematisk prosess

### Forstudie/målanalyse

- o Kost/nytte, risikoanalyser, hvorfor skal vi lage et system?

### Kravinnsamling og -analyse

- o hva ønsker/trenger interessentene? Prioritering av kravene.

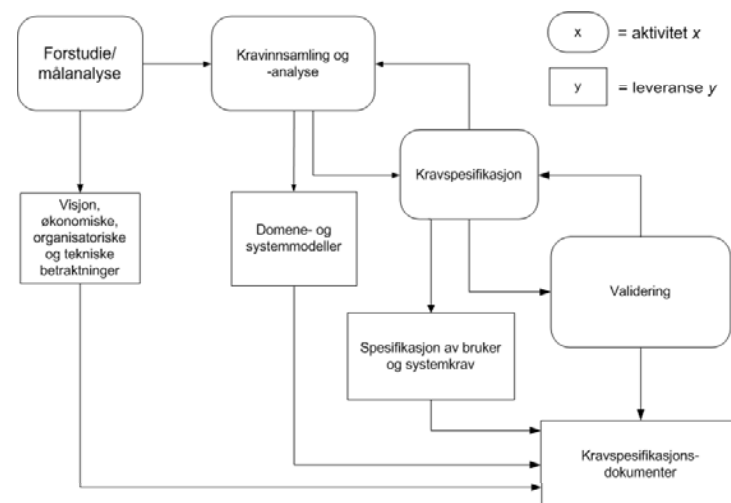
### Spesifiser kravene

- o utgangspunkt for kontrakt mellom kunde og leverandør
- o utgangspunkt for kostnadsestimater
- o utgangspunkt for design, implementasjon og testing

### Valider spesifikasjonen

- o definerer kravspesifikasjonen det interessentene faktisk vil ha?

## Overordnet kravhåndteringsprosess



Basert på Sommerville: "Software Engineering" – s143

### Requirements Document Table of Contents

1. Project Preliminaries
    - 1.1 Purpose and Scope of the Product
    - 1.2 Business Context
    - 1.3 Stakeholders
    - 1.4 Ideas for Solutions
    - 1.5 Document Overview
  2. System Services
    - 2.1 The Scope of the System
    - 2.2 Function Requirements
    - 2.3 Data Requirements
  3. System Constraints
    - 3.1 Interface Requirements
    - 3.2 Performance Requirements
    - 3.3 Security Requirements
    - 3.4 Operational Requirements
    - 3.5 Political and Legal Requirements
    - 3.6 Other Constraints
  4. Project Matters
    - 4.1 Open Issues
    - 4.2 Preliminary Schedule
    - 4.3 Preliminary Budget
- Appendices
- Glossary
  - Business Documents and Forms
  - References

Maciaszek s.104

## Fase 1: Forstudie/målanalyse

- Engelsk: Inception, solution envisioning
- Idefasen (Inception) i Unified Process (Hasle 6.3), "Behovsfasen" i PS2000 (neste forelesning)
  - o Analyser nåsituasjonen, ønsket situasjon og mulige tiltak for å oppnå ønsket situasjon
  - o Hvilke (del)mål kan oppnås ved å lage et nytt IT-system? Hvem er interessentene?
  - o Hva er kost/nytte for forskjellige delmål? Risikomomenter?
  - o Kan systemet integreres med andre systemer som allerede er i bruk?
  - o Prosjektmandat: "Ja, vi skal lage et system for å oppnå følgende mål"
- Se også Hasle 1-3, Maciaszek 2.1

## Fase 2: Kravinsamling og -analyse

- ❑ **Engelsk: Requirements Elicitation, requirement collection, capture or discovery**
- ❑ **Identifiser interessentenes krav, prioriter og løs konflikter mellom krav, valider kravene**
  - Omfatter mange av de samme aktivitetene som i foranalysen, bortsett fra at man nå typisk har et prosjektmandat og derfor innhenter flere fakta og systematiserer dem.
- ❑ **Alle interessenter bør involveres!**

{Husk: Interessent = en eller annen som har en direkte eller indirekte interesse av at systemet utvikles}

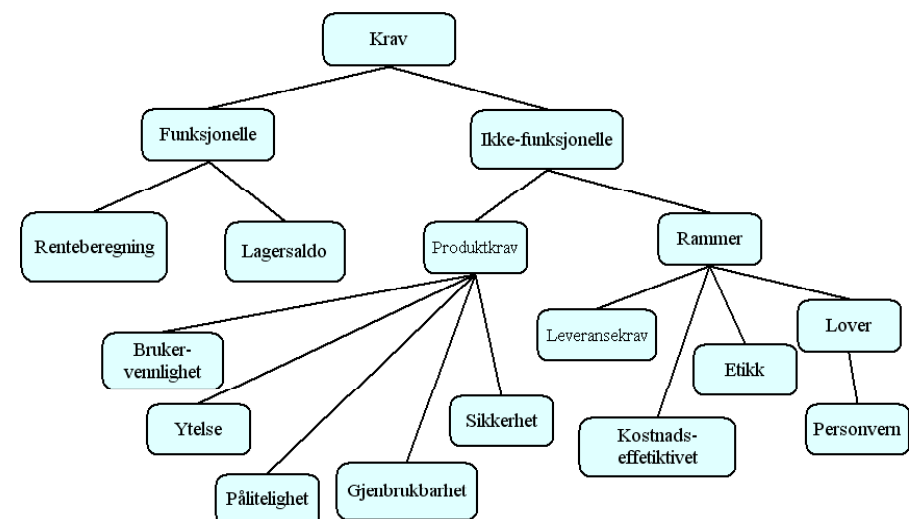
## Fase 3: Kravspesifikasjon

- ❑ **Spesifiser kravene mest mulig presist, i et passende format, for eksempel ved bruk av modelleringsnotasjoner som Unified Modeling Language (UML).**
  - Detaljer kommer i senere forelesninger!
- ❑ **Kravspesifikasjonen er et utgangspunkt for å analysere *hvordan* systemet skal oppfylle kravene, men omfatter ikke implementasjonsdetaljer**
- ❑ **Presis spesifikasjon av omfanget på systemet**
  - UML: Aktører (actors) og bruksmønstre (use cases)
- ❑ **Leveransen fra en spesifikasjonsaktivitet er et utvidet kravdokument, som vi gjerne kaller et kravspesifikasjonsdokument (eller en kravspec)**

## Fase 4: Validering av kravspesifikasjonen

- ❑ **Forståelighet – forstår interessentene kravene?**
  - For kompliserte – kan kravet deles opp mindre bestanddeler?
  - Tvedydighet – har kravet flere mulige tolkinger?
- ❑ **Konsistens – inneholder kravene selvmotsettelser?**
- ❑ **Verifiserbarhet (testbarhet) – klarer du å teste om kravene er oppfylt?**
  - Hvis ikke, så er det ikke formulert presist nok
- ❑ **Sporbarhet – hva/hvem er kilden til kravet?**
  - Viktig når endringer må gjøres eller når man må prioritere vekk krav
- ❑ **Endringsevne – hva er konsekvensene av å endre et krav?**
  - Påvirker det andre krav?
- ❑ **Kompletthet – mangler det krav?**
- ❑ **Unødvendige krav – ligger kravene innenfor prosjektmandatet?**
- ❑ **Realisme – er kravene realistiske, gitt tilgjengelig ressurser?**
- ❑ **For tidlig design – er kravet et skjult designelement/føringer på implementasjonen?**

## Funksjonelle og ikke-funksjonelle krav



## Ikke-funksjonelle krav – eksempler (1)

- ❑ Brukervennlighet – er systemet lett å lære/bruke?
- ❑ “Brukervennlighet” er avhengig av krav til opplæring og varierer for forskjellige brukergrupper
- ❑ Vurder forskjellige måter som brukervennlighet kan måles:
  - Hvor lang tid tar det å lære systemet for nybegynnere?
  - Hvor mange “brukerfeil” oppstår med erfarne brukere?
  - Hvor ofte får brukerne meningsløse tilbakemeldinger?
  - Hvor mange valg har hjelpefunksjoner?
- ❑ Spesielt kritisk for web-baserte applikasjoner med store, heterogene brukergrupper



## Ikke-funksjonelle krav – eksempler (2)



### Pålitelighet

- Feilrater (mean time between failures (MTBF))
- Oppetid (% tid tilgjengelig for bruker)



### Ytelse – kan måles på forskjellige måter

- Kapasitet (transaksjoner pr. time, jf. Julehandel for BBS)
- Responstid (min/maks/gjennomsnitt)
- Antall samtidige brukere



### Sikkerhet – f.eks. Grad av datakryptering og valg av autentiseringsprotokoller/innlogging

- Eget lfi-kurs om sikkerhet planlegges

## Ikke-funksjonelle krav – eksempler (3)

Eksemplene så langt beskrev ønskede egenskaper ved produktet. Andre ikke-funksjonelle krav kan beskrive *begrensninger eller rammer*. For eksempel:



Kostnader og ressurser er alltid en begrensning!



Leveransetidspunkt (påvirker også kostnader og ressursbruk)



Krav om samvirke med andre systemer



Gjenbruk av eksisterende teknologi – programmeringsspråk, verktøy, komponenter



Lover - personvern

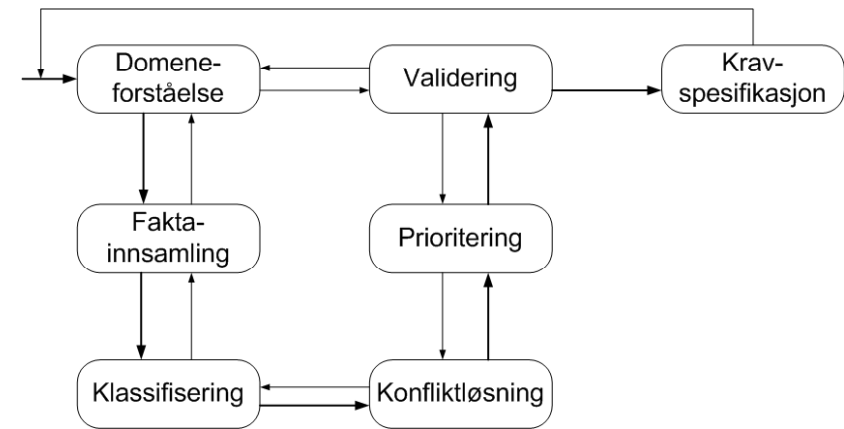
## Kravinnsamling – utfordringer

- ❑ Forskjellige forretningsområder har ofte sin egen terminologi
- ❑ Forskjellige organisasjoner har egen terminologi, struktur og forretningsprosesser, som en utvikler kanskje ikke kjenner til.
- ❑ Interessenter vet ikke nøyaktig hva de vil ha eller kjenner ikke til tekniske muligheter og begrensninger
- ❑ Motstridende krav fra forskjellige interessenter, forskjellige meninger om hva som er viktig
- ❑ Ustabile organisasjoner (reorganisering, oppkjøp)

## Prosess for kravinnsamling og -analyse

- ❑ Forstå domenet – forretningsområde og terminologi
- ❑ Faktainnsamling – identifiser krav ved å intervju interessenter, observere prosesser, analysere dokumenter, osv.
- ❑ Klassifisering – organiser kravene i hierarkier eller grupper
- ❑ Konfliktløsning – identifiser og løs potensielle konflikter mellom krav fra forskjellige interessenter
- ❑ Prioritering – identifiser de viktigste kravene
- ❑ Validering – komplett, konsistens, realisme

## Prosess for kravinnsamling og -analyse (2)



## Tradisjonelle metoder for faktainnsamling (Emne 3 i Hasle)

- ❑ Intervjuer
- ❑ Spørreskjemaer
- ❑ Observasjon
- ❑ Studere dokumenter og eksisterende systemer

## Intervjuer



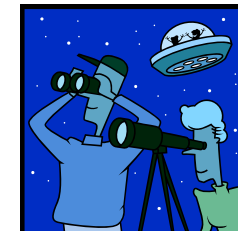
- ❑ Typer intervjuer
  - Ustrukturerede intervjuer/samtaler
  - Strukturerede intervjuer med både åpne spørsmål og forhåndsdefinerte spørsmål
  - Semi-strukturerede intervjuer: En blanding!
- ❑ Typer spørsmål
  - Om eksisterende prosesser, framtidvisjoner, alternative ideer, minimale, akseptable og optimale løsninger, spesifikke detaljer, andre informasjonskilder
- ❑ Datainnsamling
  - 📝 notater
  - 🎧 opptak (men NB! mikrofonskrekk!) + transkribering (tidkrevende!)
- ❑ Send gjerne en kort rapport til intervjuobjektet innen et par dager slik at hun kan rydde opp i evt. feil og misforståelser

## Spørreskjemaer



- ❑ Gjerne i etterkant av noen intervjuer
- ❑ Fordeler
  - Respondenter har tid til å forberede seg!
  - Relativt sett mindre tidkrevende enn intervjuer
- ❑ Ulemper
  - Vanskelig å formulere presise spørsmål, krever ofte gode pilotstudier
  - Ingen mulighet til å presisere uklare spørsmål og svar
  - Hvem gidder å svare?? Skjevt utvalg...
- ❑ Typer spørsmål
  - Flervalg eller åpne
  - Prioritering av forskjellige alternativer, kartlegging av arbeidsoppgaver

## Observasjon



- ❑ I tillegg til intervjuer og evt. spørreskjemaer
- ❑ Observere arbeidsprosesser
- ❑ Tre hovedformer
  - Passiv observasjon (flua på veggen)
  - Aktiv observasjon (hva gjør du hvis ...?)
  - Egenobservasjon, spesielt for kunnskapsintensivt arbeid
- ❑ Ikke alle oppfører seg “normalt” når de blir observert

## Studere eksisterende dokumenter og systemer

- ❑ Dokumenter som utveksles i organisasjonen
  - Prosedyrer, avtaler, planer, strategier, retningslinjer, rapporter og formularer
- ❑ Standarder, reglementer og lovverk
- ❑ Dokumentasjon av eksisterende systemer
  - De fleste systemer samvirker med andre systemer
  - Evt. feilrapporter og endringsønsker i eksisterende system!



## Andre metoder for faktainnsamling

- ❑ Prototyping (Hasle 4.4)
  - Bruk-og-kast prototyper for å teste ut ideer og vise muligheter til brukerne
  - GENOVA kan lage prototyper basert på kravspesifikasjonen
- ❑ Idédugnad (Maciaszek 2.2.3.2)
  - Uformelt møte for å generere ideer og identifisere mulige løsninger på problemer.
  - Deltakere: Gjerne 12-20 som har samme status i møtet
  - Moderator definerer problemområdet og holder diskusjonene rettet mot dette.
  - Ideer analyseres og prioriteres (for eksempel "gule lapper")
- ❑ Joint Application Development – JAD (Hasle 7.2)
  - Strukturert møteopplegg som minner en del om en idédugnad.
  - Deltakere: Møteleder (god domenekunnskap, beslutningstaker), Sekretær (sterk systemutviklingskompetanse), Kunder og sluttbrukere (hoveddeltakere), utviklere (observatører)
  - Utdanner sluttbrukere og utviklere, som raskere oppnår en felles forståelse for kravene

## Klassifisering av krav



- ❑ Vi må ha en måte å klassifisere krav for å identifisere konflikter og redundante krav
- ❑ Kotonya & Sommerville:
  - Gruppering – Lag grupper av krav som “hører sammen” (høy kohesjon) med få avhengigheter til andre grupper (lav kobling)
  - Abstraksjon – prøv å finn felleselementer i kravene som kan deles mellom forskjellige funksjoner eller interessenter (hovedfunksjonalitet)
  - Prosjeksjon – betrakt systemet fra forskjellige perspektiver (sykehussystem: sykepleier, lege, administrasjon), og organiser kravene rundt disse

## Konfliktløsning



- ❑ Identifiser og forsøk å løse konflikter mellom kravene til forskjellige interessenter
- ❑ Sjekk om det finnes inkonsistens mellom kravene
- ❑ Sjekk om kravene er reduntante (se også klassifisering)
- ❑ Lag en avhengighetsmatrise mellom kravene:

Krav	K1	K2	K3	K4
K1				
K2	Konflikt			
K3				
K4		Overlapper	Fortsetter	

## Prioriter kravene



- ❑ Bruk klassifisering, avhengighetsmatrise samt kost/nytte-vurderinger (ikke trivielt) for å møte budsjett eller for å definere delleveranser

Delleveranse	Utbytte	Innsats	Utbytte/innsats
Renteberegning	25	5	5
Lagersaldo	9	0,5	18
Bidragsregistrering	5	7,5	0.67

## Endringshåndtering av krav

- ❑ Krav kan endres, slettes, eller nye krav kan oppstå
- ❑ Endringshåndtering betyr at man må
  - dokumentere endringsforespørsler,
  - gjøre en konsekvensanalyse,
  - Implementere endringen
- ❑ Endringsforespørsler lagres og spores gjerne i et eget “change management” verktøy
- ❑ Implementasjonen av endringen kan spores i et konfigurasjonsstyringsverktøy.



# Samling av trådene



## Systemutvikling som helhet

1. Systemutvikling: motivasjon ..... Jo Hannay, Simula & Ifi
2. Systemutviklingsprosessen ..... Rune Steinberg, Visma Software AS
3. Prosjektledelse og prosjektarbeid ... Rune Steinberg, Visma Software AS

## Kunde/leverandør/bruker-forhold

4. Kravhåndtering ..... Erik Arisholm, Simula & Ifi
5. Avtaler & kontrakter ... Jørgen Petersen, Promis AS
6. Estimering ..... Stein Grimstad, Simula
7. Jus & etikk ..... Dag W. Schartum

## Systemets struktur og design

8. Modellering av krav med use cases ... Erik Arisholm, Simula & Ifi
9. Objektorientert analyse (2 forel.) ..... Erik Arisholm, Simula & Ifi
10. Persistens og databaser ..... Erik Arisholm, Simula & Ifi

## Koding, validering og vedlikehold

Dag Lorås, Visma Software AS

12. Modellbasert utvikling med Genova ... Esito AS
13. Validering og verifisering (2 forel.) ..... Lionel Briand, Simula & Ifi
14. Konfigurasjonsstyring ..... Hans Christian Benestad, Simula

# 404 NOT FOUND

