

INF1030: Systemer, krav og konsekvenser

20. mars 2025

Kravhåndtering, Lærebok kap. 4



Professor Dag Sjøberg

Plan

- Generelt om krav
- Funksjonelle krav
- Ikke-funksjonelle krav
- Kravspesifikasjoner
- Kravhåndteringsprosessen

Kravhåndtering

- Hensikten med å utvikle eller forbedre et IT-system:
 - å løse utfordringer eller
 - utnytte potensialer
- Kravhåndtering er prosessen å identifisere, analysere og spesifisere kravene til det nye eller forbedrede systemet

Hensikten med å lage en kravspesifikasjon

- Basis for anbud
 - ulike tilbydere vil kunne tilby ulike måter å løse kundens behov på
- Basis for kontrakt
- Basis for design og implementasjon av systemet

Typer krav

- Brukerkrav
 - Krav uttrykt i naturlig språk eller diagrammer som viser ønskede tjenester (funksjoner) til systemet og føringer som gjelder (andre kvalitetsegenskaper)
 - Skal forstås greit av kunden
- Systemkrav
 - Strukturert, detaljert beskrivelse av systemets funksjoner og kvalitetsegenskaper
 - Definerer mer detaljert hva som skal designes og implementeres
 - Utgangspunkt for kontrakt mellom kunde (oppdragsgiver) og utviklerorganisasjon

Plan

- Generelt om krav
- Funksjonelle krav
- Ikke-funksjonelle krav
- Kravspesifikasjoner
- Kravhåndteringsprosessen

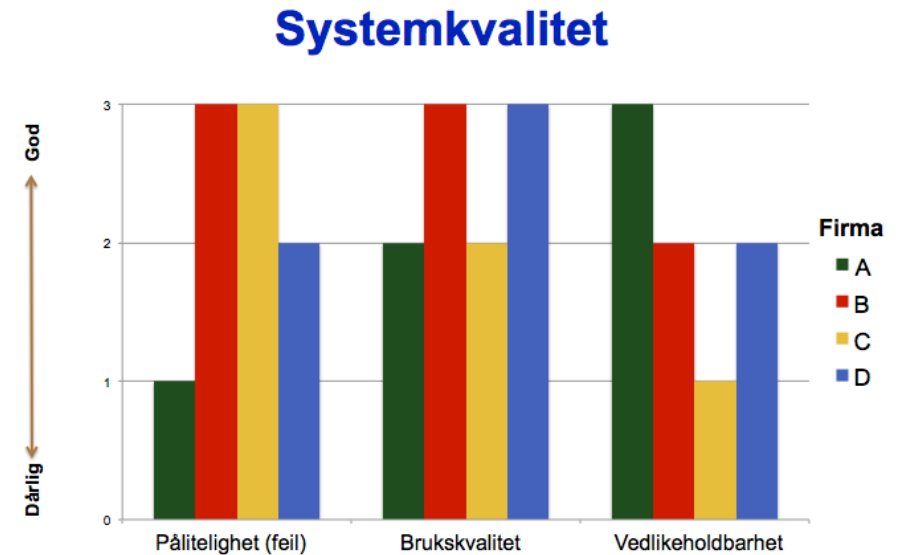
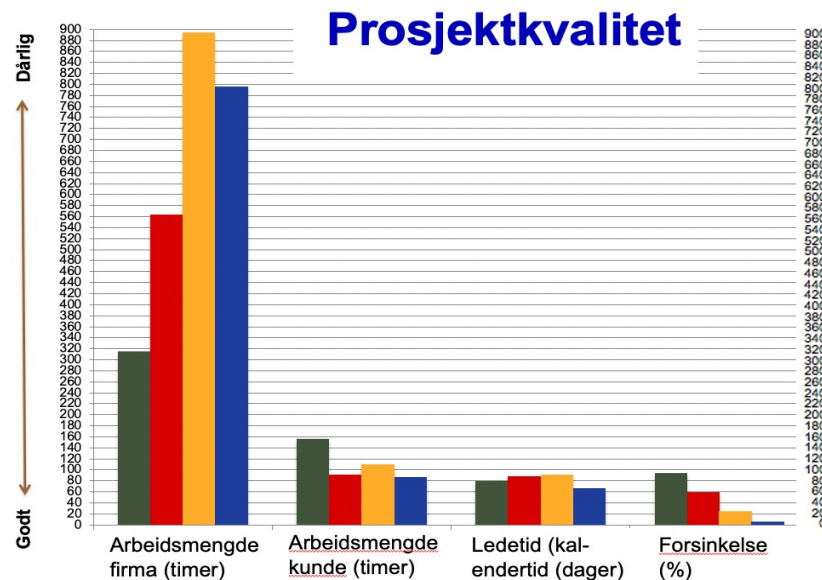
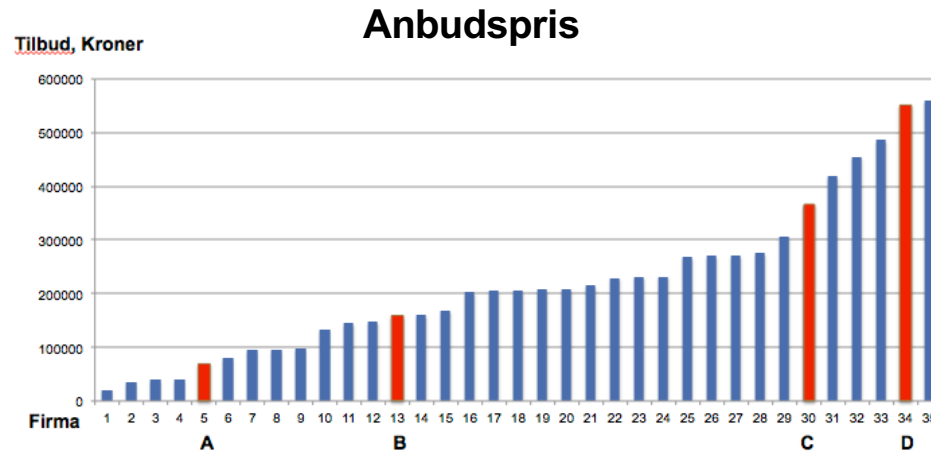
Funksjonelle krav

Hva systemet skal gjøre

- Hvilke tjenester (funksjoner) skal systemet tilby?
 - Tjenestededesign en aktivitet for å forstå hva brukerne trenger
 - Resultatet av tjenestededesign er input til kravspesifisering
- Hvordan skal det reagere på ulike typer input?
- For å avgrense systemet, vil man også kunne beskrive hva systemet *ikke* skal gjøre

Eksempel 1:

Database over Empiriske Studier (DES)



Overordnede funksjonelle krav i DES

- “DES should support the processes related to storing and reporting Studies
- DES should enable internal and external researchers to finding information about Studies
- DES should connect information about
 - Responsible for the Studies to the *Employee database* and
 - Publications from the Studies to the *Publication database*”

Detaljerte funksjonelle krav

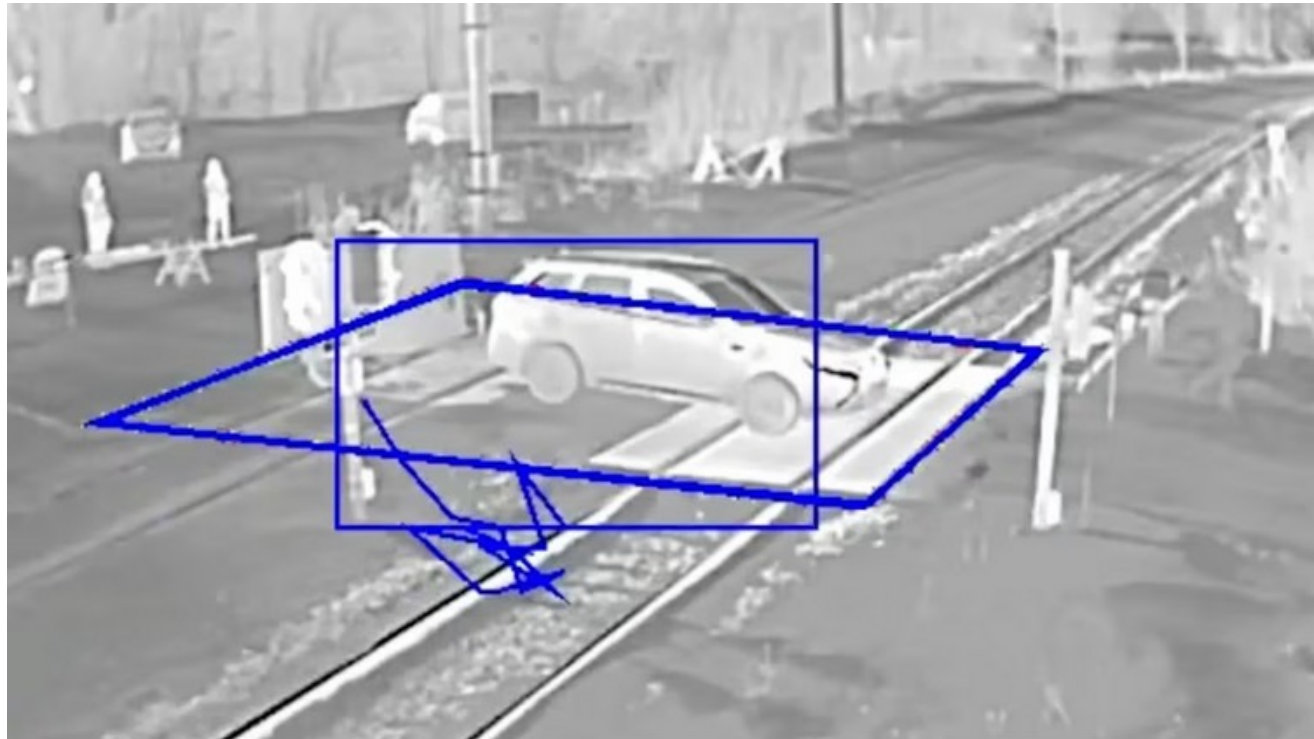
“Function: Register new Study

1. Administrator log in
2. Insert Study information
3. Control that all mandatory information is included
4. The name (Last name + First name) of the administrator is registered as *Study Owner* by the system”

Eksempel 2: Hindre togpåkjørrelse

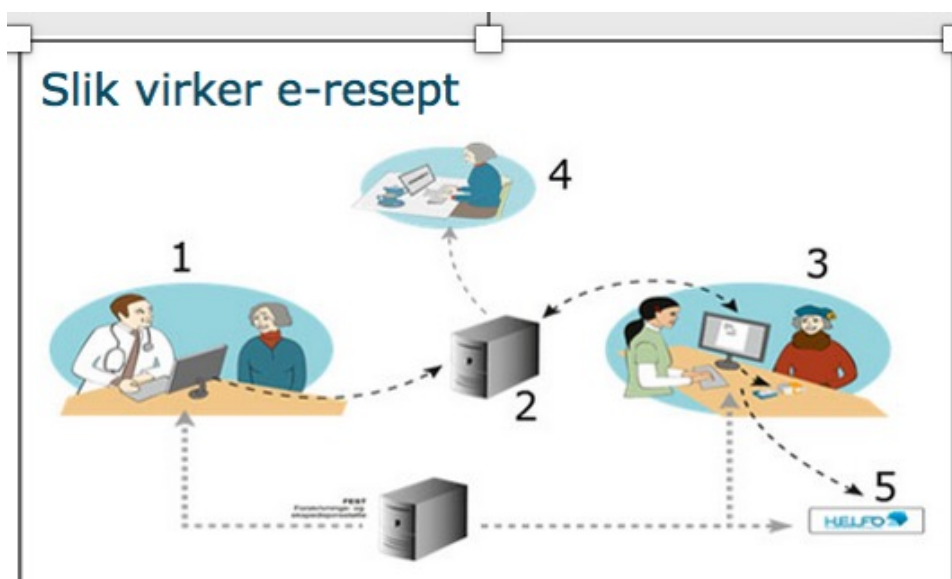
Krav: Et tog må stanse hvis det er noe på en planovergang med bommene nede

Bane NOR bruker et varslingsystem basert på kunstig intelligens (KI) for å forhindre ulykker på planoverganger



Eksempel 3: Overordnede funksjoner i E-resept

En elektronisk samhandlingskjede håndterer resepten fra forskrivning via utlevering til oppgjør og kontroll.



1. **Rekvirenten** forskriver på basis av oppdatert produktinformasjon fra Forskrivnings- og ekspedisjonsstøtte (FEST). Med pasientens samtykke kan fastlegen få informasjon om utleveringer basert på resepter fra andre rekvirenter. Alle leger kan med pasientens samtykke få se gyldige resepter som andre rekvirenter har utstedt. Det gir bedre oversikt og reduserer risikoen for feilmedisinering.
2. **Reseptformidleren** mottar resepten straks rekvirenten har sendt den. Resepten blir slettet automatisk fire uker etter at den er ferdig ekspedert. Når den utløper på dato, blir den slettet med det samme.* Lege kan tilbakekalle gyldige resepter fra Reseptformidleren, også de som andre rekvirenter har sendt. Dette sikrer at Reseptformidleren bare inneholder resepter på legemidler pasienten skal bruke.
3. **Apoteket eller bandasjisten** søker fram resepten i Reseptformidleren når kunden oppgir fødselsnummer eller navn og fødselsdato. De kan også be kunden legitimere seg.
4. **Pasienten** kan administrere e-reseptene sine med tjenesten [Mine resepter](#). Pasienten kan også holde oversikt ved å be om utskrift hos legen eller på apoteket. Resepten ligger trygt i Reseptformidleren og kan ekspederes hvor som helst i Norge. Fastlegen bør gi pasienten en utskrift av oppdatert liste over Legemidler i bruk.
5. **Helseøkonomiforvaltningen (Helfo)** mottar apotekenes og bandasjistenes krav om refusjon for varer på blå resept. Helfo mottar også elektroniske søknader om individuell refusjon.

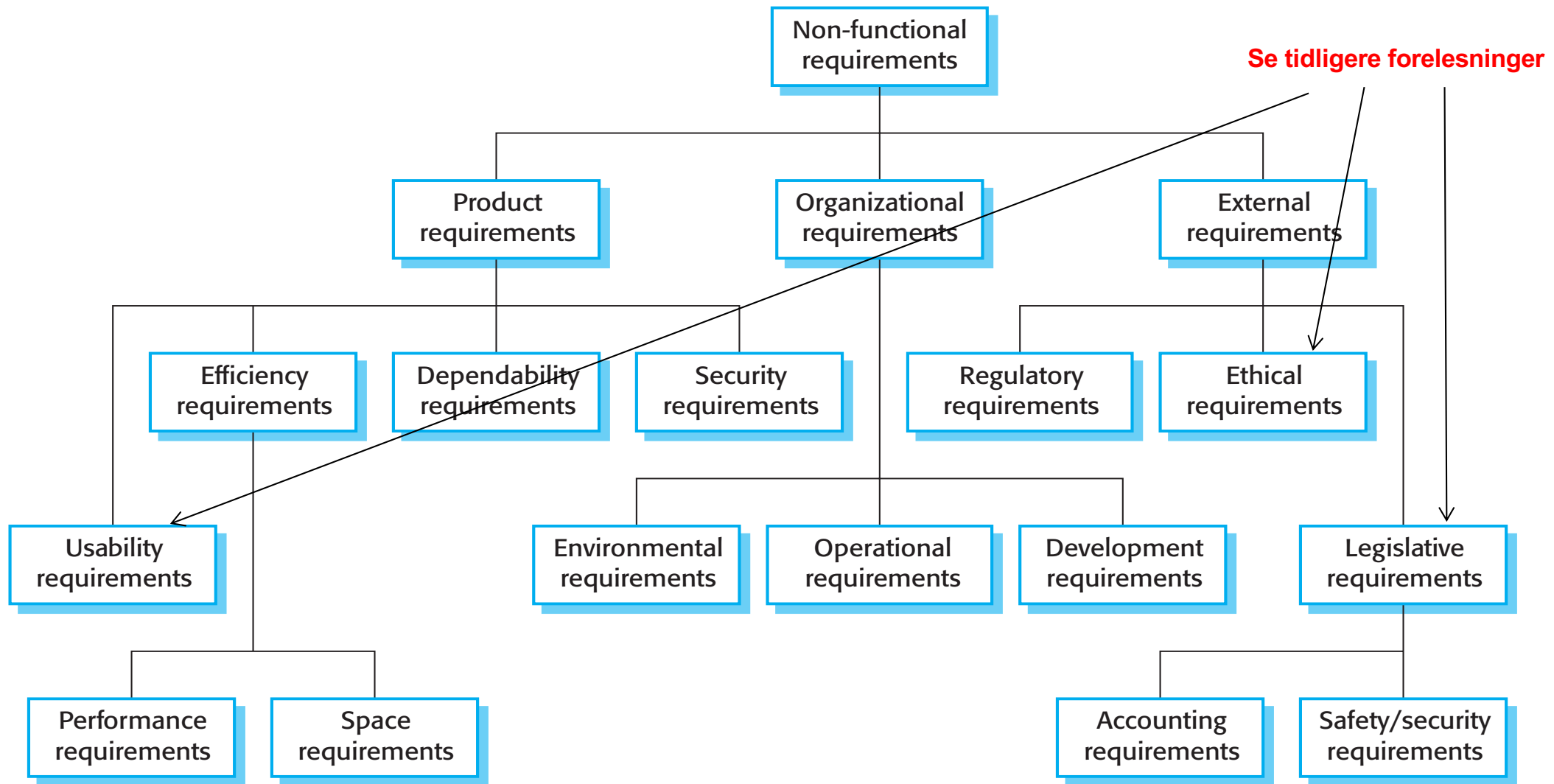
Plan

- Generelt om krav
- Funksjonelle krav
- Ikke-funksjonelle krav
- Kravspesifikasjoner
- Kravhåndteringsprosessen

Ikke-funksjonelle krav

- *Hvordan* systemet skal implementere de funksjonelle kravene

Mange typer ikke-funksjonelle krav



Produktkrav: "Usability requirements"

– er systemet lett å lære og bruke?

- Brukskvalitet/Brukervennlighet
 - avhenger av krav til opplæring
 - varierer for ulike brukergrupper
 - universell utforming
- 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?
 - Responstid: Tid fra brukeren trykker 'OK' til systemet svarer

Produktkrav:

”Efficiency requirements” (Effektivitet)

- Ytelse (”Performance requirements”) – **må kvantifiseres**
 - Kapasitet (transaksjoner pr. timer i betalingskortsystemer)
 - Antall samtidige brukere
 - Responstid (min/maks/gjennomsnitt ved ulik belastning)
- Lagringsplass (”Space requirements”)

Responstid i E-resept Versjon 2.0

4.4 Krav til ytelse

NB! Ikke-funksjonelt krav

I tabellen under er det beskrevet responstidskrav. Disse er knyttet til maks responstid for sekvensen av meldinger som sendes. Prosjektet vil håndtere følgende krav til ytelse for grensesnitt og funksjoner:

Legepraksis med 4 leger			
Melding inn	Volum per dag	Responstid	Hendelse/kommentar
Utleveringsrapport (M6 / M8)	0 – 250	NA	
Varer i bruk M25.2, M25.3	0-100	NA	
Applikasjonskvittering (M1, M2, M5 og M25 – smtp)	0 – 250	5 sek	Gjelder responstid fra EPJ-system sender melding til Reseptformidleren til det mottar svar fra Reseptformidleren
M1 (WS)	0-2000	<3 sek	
M5 (WS)	0-500	<3 sek	
M25.1 (WS)	0-20	<3 sek	
Svar på søknad til Legemiddelverket (M15)	0 – 5	NA	
Svar på spørsmål om resepter i Reseptformidleren (M9.5/M9.6)	0 – 500	< 3 sek	Gjelder responstid fra EPJ-system sender forespørsel til det presenterer svar for rekvirent
Svar på spørsmål om varer i bruk (M9.11-M9.12)	0-5	<3 sek	Gjelder responstid fra EPJ-system sender forespørsel til det presenterer svar for rekvirent

Versjon 2.05

- mer detaljert
- større krav til volum
- 38 sider kortere dokument (bl.a. mindre repetisjon)

4.4 → Krav til ytelse

I tabellen under er det beskrevet responstidskrav. Disse er knyttet til maks responstid for sekvensen av meldinger som sendes. Aktørene vil håndtere følgende krav til ytelse for grensesnitt og funksjoner:

Legepraksis med 4 leger			
Melding	Volum per dag	Responstid	Hendelse/kommentar
Utleveringsrapport (M6 / M8)	0 – 500	NA	
Legemidler i bruk M25.2, M25.3	0-100	NA	
Sending resept M1 (WS)	0-2000	<3 sek	
Sending tilbakekalling M5 (WS)	0-1000	<3 sek	
Sending LIB M25.1 (WS)	0-20	<3 sek	
Søknad til HELFO	0 – 5	NA	
Svar på søknad til Legemiddelverket (M15)	0 – 5	NA	
Spørsmål om resepter i Reseptformidleren	0 – 500	<3 sek	
Spørsmål om Legemidler i bruk	0-500	<3 sek	
Henting av referansenummer	0 – 1	<3 sek	
Henting av FEST	0-1	<5 sek	

Sykehus med poliklinikk og utskrivning av pasienter			
Melding inn	Volum per dag	Responstid	Hendelse/kommentar
Utleveringsrapport (M6 / M8)	0 – 1000	NA	
Sending resept M1 (WS)	200-20000	<3 sek	
Sending tilbakekalling M5 (WS)	10-5000	<3 sek	
Svar på søknad til Legemiddelverket (M15)	0 – 50	NA	
Spørsmål om resepter i Reseptformidleren	0 – 10000	<3 sek	
Spørsmål om Legemidler i bruk	0-10000	<3 sek	
Henting av referansenummer	0 – 1	<3 sek	
Henting av FEST	0-1	<5 sek	

Ulike versjoner av E-resept sin kravspesifikasjon

Dokumenthistorikk

<i>Dato</i>	<i>Versjon</i>	<i>Laget av</i>	<i>Godkjent av</i>	<i>Endring</i>
24.2.10	1.8			Første versjon i produksjon
6.4.10	1.9			Innarbeidet EO51: Åpne saker 688, 705, 717, 739 og 759 Innarbeidet EO52, Åpne saker 635, 749, 753, 779, 781, 784, 785b, 792 og 799
18.03.11	2.0			Innarbeidet EO53: Åpen sak 815 Innarbeidet EO54: Åpne saker 811, 835 Diverse rettelser
15.06.11	2.0			Oppdatert etter innspill fra prosjekten
05.12.11	2.01	gusas		Endringer kap. 9.5, virkestofforskriving
04.05.12	2.02	jba		Endringer knyttet til at M25 skal gå via Reseptformidleren kap 2.3.1, 3.4.2, 4.2.2, 4.3.7, 4.4, 5.1, 5.2.6, 5.3, 5.6.1, 6.2.4, 6.2.5.6
12.02.13	2.03	gusas		Endringer utleverer kap3.2, utskrift ref.nr, og papirresepter i M10. Bruk av merknadsfelt i M10/M8.
04.04.13	2.04	gusas		V 2.5 til forankring hos aktørene
16.04.13	2.05	gusas		Er vedtatt i e-resept endringsrådsmøtet 2/5-2013, etter verifiseringsrunde.
30.09.13	2.06	OMW		Innarbeidet endringer knyttet til endringsrådsmøte 09.10.13.
12.05.14	2.07	jba		EO 196: 3.1.1.1, 3.1.1.9, 3.1.4, 4.3.1. EO 1731: 3.1.4, 5.2.3.1, 5.3.1.8. EO 5166: 2.3.1, 2.4, 3.1.1.7, 4.3.1, 5.1.1, 5.2.1.6, 5.3.1.6 EO 5482: 3.4.2.11. EO 5487: 3.4.2.1, 4.3.7, 5.2.6, 5.3.1.15, 5.3.2.8, 5.3.1.13. EO 5526: 4.3.7, 5.2.6, 5.3.13, 5.3.1.15, 5.3.2.8, 5.3.2.10-11, 6.2.4. EO 5658: 5.3.2.3. EO 5750: 5.3.2.9. EO 5751: 5.3.1.13, 5.3.1.15, 5.3.2.8, 5.3.2.10. EO 5782: 9.2.1. EO 5783: 2.3.1, 3.4.2.7, 5.1.1, 5.2.6.7, 5.3.2.12, 5.6.1, 6.2.4.5 EO 5940 5.3.2.11 EO 6010: 4.2.7, 4.3.4, 5.2.3, 5.2.4.1, 5.2.4.4, 5.3.1, 5.3.2, 8.2.6. EO 6019 3.4.2.10, 5.3.1.13
03.11.15	2.08	jba		Feilretting. Alle EOer behandlet i endringsrådet i perioden juni 2014 – mai 2015 unntatt EO 4487 og EO 7761. EO 8841. Innarbeidet LIB og multidose, Krav og konvensjoner august 2014. Innarbeidet Magistrelle resepter i e-resept, Krav og modellerer, juni 2014
14.03.18	2.09	alien		Alle Ers-saker besluttet i Endringsråd i perioden juni 2015 – mars 2018
24.08.18	2.10	ksm		Innarbeidelse av PLL-funksjonalitet. Endret fra LIB til PLL gjennomgående. Feilrettinger.

Produktkrav:

”dependability requirements”

– I hvilken grad kan man stole på systemet?

- Pålitelighet/feil (”reliability”) – **må kvantifiseres**
 - Feilrater (mean time between failures (MTBF))
 - Oppetid (% tid tilgjengelig for bruker)
- Ulike systemer, ulike krav

Oppetid i E-resept

Detaljert funksjonell spesifikasjon eResept
Versjon 2.0

Side 125 av 204

Likt i v2.10

5.7 Drift og brukerstøtte

5.7.1 Drift

Reseptformidleren skal ha en tilgjengelighet på 99,99 %. Tilgjengelighet på systemet basert på 24 timers drift 7 dager i uken, 365 dager i året. Kravet testes ved å måle over en periode på en uke og tilsvarer mindre enn ett minutt nedetid i løpet av en uke.

Hovedformålet med Mine resepter er å oppfylle lovens krav om pasienters rett til innsyn i informasjon som er lagret om dem. Oppetidskrav for Mine resepter er satt til 99,8 % basert på 24 timer, 7 dager i uken. Kravet testes ved å måle over en periode på en uke og tilsvarer mindre enn 20 minutter nedetid løpet av en uke.

dager i året?

6.4.1 Oppetider

Apoteket vil kommunisere med Reseptformidleren og HELFO via Apotekenes Intranett. Linjen mellom Apotekenes Intranett og apotekene skal være sikret med redundante løsninger på ulike teknologier som ivaretar en høy grad av oppetid på denne linjen. Videre kommunikasjon med Reseptformidleren går via en fastlinje fra Apotekenes intranett til NHN dedikert til dette formål. Dersom fastlinjen fra Apotekenes Intranett går ned skal det finnes en alternativ løsning for kommunikasjon basert på annen kommunikasjon via NHN. Krav om 99.99 % oppetid på Reseptformidleren i perioden 8.00-21.00 Krav om 99,9 % oppetid i perioden 21.00-8.00

Produktkrav: "Security requirements"

- Sikring av data, for eksempel
 - grad av kryptering
 - valg av autentiseringsprotokoller/innlogging



Begrepene *security and safety* brukt innen systemutvikling

- Security: IT-systemer skal hindre at de selv eller deres data blir angrepet utenfra (sikkerhet mot uønskede hendelser som resultat av overlegg)
 - Hacking
- Safety: Det skal ikke være risikabelt å bruke IT-systemer (sikkerhet mot uønskede hendelser som resultat av tilfeldigheter og ulykker)
 - Selvstyrte kjøretøy?

Sikkerhetskrav (safety) i japansk togkontroll (ATC):

- Når en datamaskin i toget mottar varsel om jordskjelv, skal bremsene settes på *innen 2 sekunder* (nytt krav, tidligere 3 sek.)

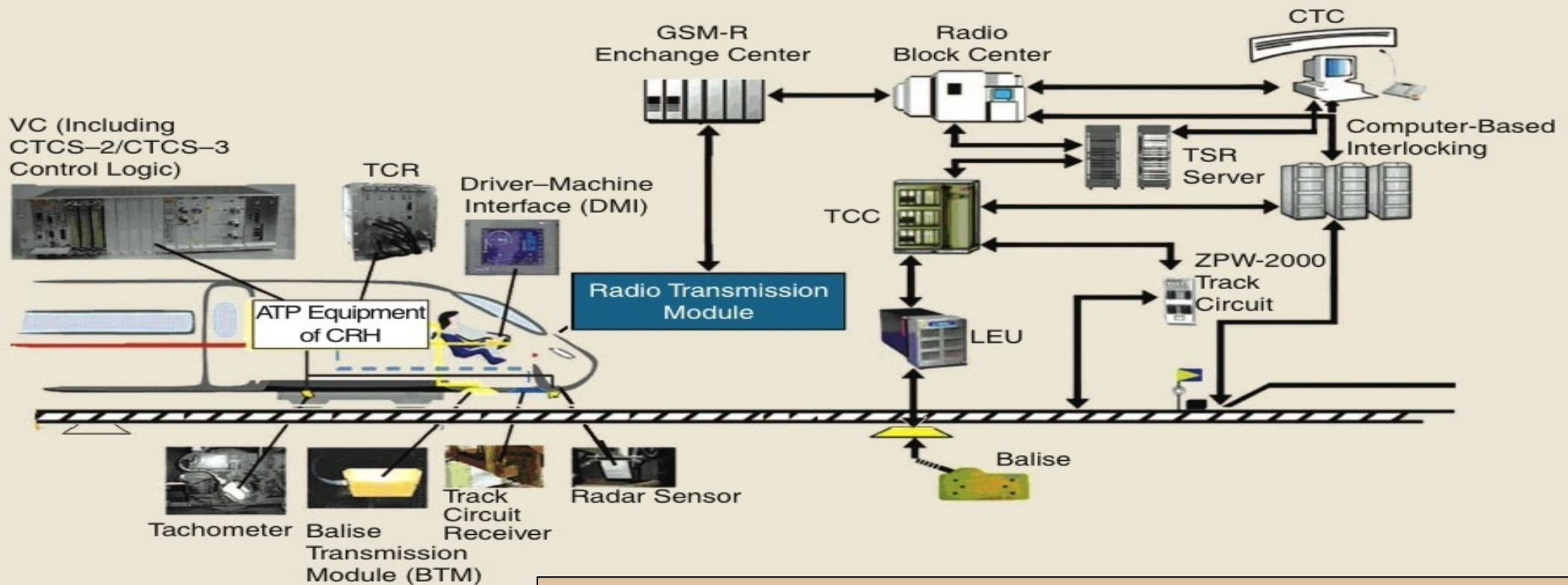


Figure 4. Structure of CTCS level-3.

Hairong Dong, Bin Ning, Baigen Cai, Zhongsheng Hou, Automatic Train Control System Development and Simulation for High-Speed Railways, *Circuits and Systems Magazine, IEEE*, 10 (2): 6–18, 2010

About the Shinkansen

- ▶ Outline
- ▶ **Safety**
- ▶ Punctuality
- ▶ High Speed
- ▶ High Frequency and Large Capacity
- ▶ Comfort



> SCMAGLEV and Railway Park

About the Shinkansen

Safety



0

fatality / injury
since 1964

- No accidents resulting in fatalities or injuries to passengers onboard since operations commenced in 1964.
- Personnel are highly-skilled in safety awareness through comprehensive training
- Train control system with sophisticated technology; continuous safety-related investment (including countermeasures for derailment and deviation for further anti-earthquake measures) in addition to reinforcement of infrastructures

Krav til pålitelighet: Høyhastighetstog i Japan

- Tokyo-Osaka (500 km): 342 tog med 424 000 passasjerer hver dag
- Gjennomsnittlig forsinkelse: 0,9 minutter
- Mesteparten av forsinkelsene skyldes "natural disasters such as earthquakes, strong winds, heavy snows and typhoons."



Organisasjonskrav: “Development requirements”



- Kostnader og ressurser er alltid en begrensning!



- Leveransetidspunkt (påvirker også kostnader og ressursbruk)



- Prosessmodeller og utviklingsmetoder
- Programmeringsspråk, verktøy, komponenter



- Standarder og regler i organisasjonen

Eksempel ikke-funksjonelle krav: “Database over Empiriske Studier”

- ***“Language***
 - Screen content, messages, database field, table, report names, online and offline documentation should all be written in UK English.
- ***Maintenance Requirements***
 - DES should use standard scripting language and HTML and standard SQL statements, including ANSI SQL 99 syntax supported by MySQL v. 3.23, to minimize the need for maintenance of the code due to new browser and/or MySQL versions.
- ***Technical documentation***
 - The code should be documented in such a way that a developer will be able to understand and maintain the code without difficulties.
- ***User documentation***
 - The use of the DES should be self explanatory. Therefore, no training or offline user documentation should be necessary. However, there should be an opening page with information when administrating the studies”.

Hvordan sjekke at krav er oppnådd?

- Ikke-funksjonelle krav ofte vanskelige å uttrykke presist, og upresise krav er vanskelige å verifisere
- Ønske hos kunden (DES-systemet)
 - "The code should be documented in such a way that a developer will be able to understand and maintain the code without difficulties"
- Verifiserbart ikke-funksjonelt krav
 - En påstand som uttrykker noe målbart som kan testes objektivt
 - Eks. romreservasjonssystem:
 - "90 % av brukerne skal bruke mindre enn 1 minutt på å reservere ønsket rom etter å ha brukt systemet 3 ganger (gjennomført vellykkede reserverasjoner)"

Kvantifisering av ikke-funksjonelle krav

Egenskap	Variabel (måling)
Hastighet	Antall transaksjoner/sekund Responstid Tid på oppdatering av skjermen
Størrelse	Gigabytes, use case-poeng
Enkelhet i bruk	Opplæringstid Antall hjelpebilder
Pålitelighet	Gjennomsnittlig tid til feil Sannsynlighet for utilgjengelighet Feilrate
Robusthet	Tid til oppstart etter feil % handlinger som fører til feil Sannsynlighet for ødelagte data ved feil
Flyttbarhet (portability)	% installasjonsavhengige kommandoer/setninger

Smidig vs. standarder/regler

- Bra å være smidig, dvs. justere kursen underveis ved behov, men:
- Spesielt i sikkerhetskritiske systemer bør regler/standarder/konvensjoner sjekkes for å hindre ulykker
 - Alle tog sjekkes hvert døgn, all skinnegang sjekkes hver uke, etc.
 - Bør ha samme forhold til programvare. Etter endringer, automatisk sjekking/testing (regresjonstesting), se IN3240 – «Testing av programvare»

Plan

- Generelt om krav
- Funksjonelle krav
- Ikke-funksjonelle krav
- **Kravspesifikasjoner**
- Kravhåndteringsprosessen

Kravspesifikasjonen (dokument)

- Spesifiserer bruker- og systemkrav.
- Ofte del av kontrakt for systemutviklingsprosjekt
 - Derfor bør være så komplett og presis som mulig
- Informasjonen i kravspesifikasjonen vil avhenge av type system og utviklingsprosjekt
- Ulike standarder
 - F.eks. utgitt av IEEE og ISO

Måter å skrive en kravspesifikasjon på

Notasjon	Beskrivelse
Naturlig språk	Kravene skrives som nummererte setninger på norsk, engelsk etc.
Strukturert naturlig språk	Naturlig språk men på en standard form (skjema). Hvert felt gir informasjon om ett aspekt ved kravene
Grafiske notasjoner	Grafiske modeller støttet av tekstbeskrivelser, beskriver funksjonelle krav; UML use case (bruksmønstre/brukstilfeller) og sekvensdiagrammer er vanlig å bruke
Matematiske spesifikasjoner	Notasjoner basert på matematiske begreper, eks. tilstandsmaskiner og mengder. Kan redusere flertydighet, men de fleste kunder forstår ikke formelle spesifikasjoner. De vil derfor ikke kunne sjekke at de faktisk representerer deres ønsker og vil derfor være skeptiske til bruk av slike spesifikasjoner i en kontrakt

Retningslinjer for skriving av kravspesifikasjon

- Bruk et standard format på alle krav
- Bruk “må” for absolutte krav og “bør” for ønsker
- Uthev teksten på spesielt viktige deler
- Unngå IT-sjargong
- Forklar hvorfor et krav er nødvendig

Utfordringer

- Forståelighet
 - Kravene er ofte uttrykt i ved bruk av spesiell fagterminologi som kan være uforståelige for systemutviklere
 - Tekniske krav kan være uforståelige for folk uten IT-kompetanse
 - Hvem innrømmer at de ikke forstår ting?
- Implisitt
 - Domenespesialister kan kjenne fagområdet så godt at de ikke tenker på å formulere krav eksplisitt som de tar for gitt
- En god systemutvikler har ofte god domene-kunnskap.
Industri og næringsliv etterspør ofte begge deler

Krav og design

- I teorien: krav uttrykker *hva* systemet skal gjøre, designet angir *hvordan* man skal lage det
- I praksis: kan være vanskelig å skjelne mellom krav og design
 - En systemarkitektur må designes for å strukturere kravene
 - Systemet vil kunne måtte samspille med andre systemer som igjen gir opphav til nye designkrav
 - En spesifikk arkitektur kan være et krav for å imøtekomme ikke-funksjonelle krav, for eksempel for å tilfredsstille lovgivning.

Kravspesifikasjon i smidige prosjekter

- I smidig systemutvikling er det færre detaljer i kravspesifikasjonen
- Kravene gjerne uttrykt som en liste av “brukerhistorier” kalt backlog
 - Merk: bruker trenger ikke være sluttbruker. ”Som sikkerhetsansvarlig ønsker jeg ...”
- Et scenario beskriver detaljert hvordan en bruker benytter en tjeneste (scenario = sekvens av hendelser)
- I statusmøter (sprint-slutt/”retrospective” i Scrum) evalueres backlog’en. Innholdet kan endres, dvs. “levende” kravspesifikasjon
- En Epic er en stor brukerhistorie eller ide (for stor for en sprint) og som etter hvert brytes ned til flere mindre brukerhistorier

Advarsel

- Hevdes ofte i smidig utvikling at det er bortkastet å bruke tid på å lage detaljerte kravspesifikasjoner fordi kravene endrer seg likevel
- Brukes som unnskyldning for ikke å jobbe nok med kravspesifikasjonen, spesielt bør fundamentale krav spesifiseres tidlig
- Bare bruk av user stories være en utfordring i store prosjekter. Hvordan vil kravspesifikasjonen til E-resept se ut – 2000 gule lapper?
- Man kan ikke bare sett i gang å programmer store, komplekse IT-systemer. Man er nødt til å identifisere (mange) krav på forhånd

Plan

- Generelt om krav
- Funksjonelle krav
- Ikke-funksjonelle krav
- Kravspesifikasjoner
- **Kravhåndteringsprosessen**

Kravhåndteringsprosessen

- Hvordan
 - samle inn,
 - analysere,
 - validere,
 - organisere og
 - endre kravene til et system



Se figur 4.6 i Sommerville

Aktivitet 1: Forstudie/målanalyse

- Analyser nå-situasjonen, ønsket situasjon og mulige tiltak for å oppnå ønsket situasjon
- Hvilke (del)mål kan oppnås ved å lage et nytt eller endre et IT-system?
- Hva er kost/nytte for forskjellige delmål? Risikomomenter?
- Prosjektmandat:
 - “Ja, vi skal lage et system for å oppnå følgende mål:” Eller:
 - «Vi kjøper og tilpasser et eksisterende system (hyllevare)»

www.menti.com



Hva kan være fordeler ved å utvikle et system selv versus kjøpe et ferdigutviklet system (hyllevare)?

Aktivitet 2: Kravinnsamling og -analyse

- Engelsk: Requirements elicitation, collection, capture or discovery
- Forstå domenet – forretningsområdet og terminologi
- Identifiser interessentenes krav
- Organiser kravene i hierarkier eller grupper
- Identifiser og løs konflikter mellom krav
 - 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

En prosjektleder må forholde seg til ulike interessenter (stakeholders)

Fra forelesning 23. januar:

“En interessent er en person, gruppe eller organisasjon som kan påvirke eller bli påvirket av en sak eller et anliggende, for eksempel en digital teknologi.”

- Systemeiere og forvaltere: økonomi
- Oppdragsgivere (kunder): prioriterer de eller vil de ”ha alt, og det feilfritt og med en gang”?
- Brukergrupper: brukervennlighet
- Ledere: planer, mål, ikke overraskelser
- Utviklere: god teknisk løsning, stilig
- Vedlikeholdere: feilfritt, forståelig og veldokumentert
- Andre interessenter (fagforeninger, lovgivere, andre systemer)

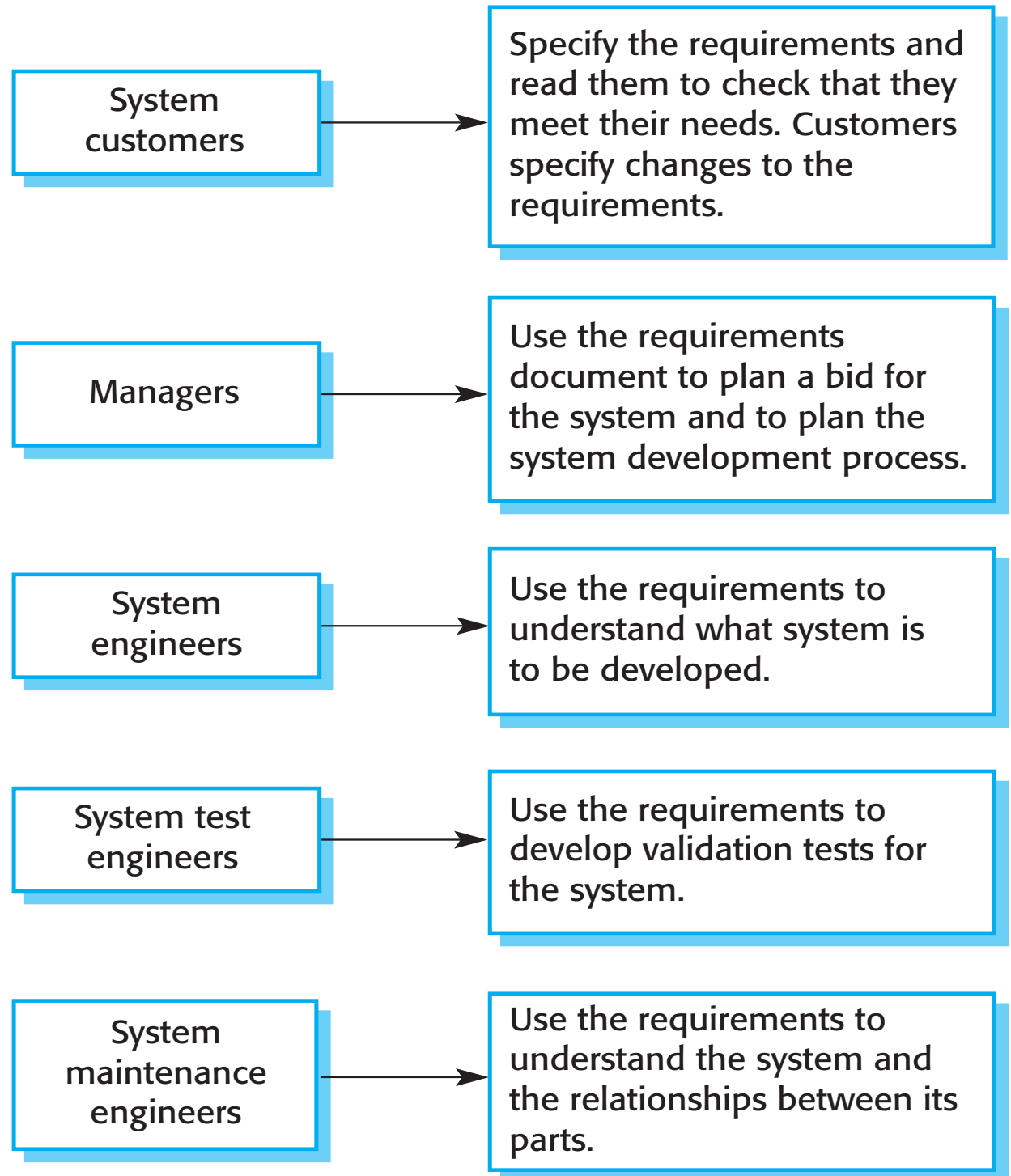
- Hvilke interessenter vil E-resept ha?

Systemeiere og forvaltere: økonomi

Det vil alltid koste noe å utvikle et IT-system

- Man vil ha et budsjett. Noen må betale hvis budsjettet overskrides
- Største utgiftspost er normalt lønn til dem som skal utvikle og vedlikeholde systemet
- I prosessen med å finne ut hva systemet skal gjøre (funksjonalitet) og hvilke andre kvaliteter (ikke-funksjonelle krav), må man prøve å kvantifisere fordelene i forhold hvor mye penger som kan brukes (kost-nytte vurdering)

Brukere av kravspesifikasjonen



Figur fra Ian Sommerville

Kravinnsamling – utfordringer

- Ulike forretningsområder har egen terminologi
- Ulike 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, meninger om hva som er viktig, organisasjonsstruktur og politikk. Det kan være skjulte agendaer
- Hvis det er umulig å oppnå konsensus, må det skjæres igjennom
- Kravene vil ofte endres underveis, nye interessenter dukker opp, forretningsområdet endrer seg, organisasjonen endrer seg (reorganisering, oppkjøp) etc.
- Må skille mellom '*need to have*' og '*nice to have*'

Hvordan prioritere krav?

- Helseplattformen Midt-Norge: over 4000 krav

Hvordan samle inn kravspesifisering-informasjon?

- Finnes en rekke metoder og teknikker
 - Bruksmønstre/brukstilfeller (use cases)
 - Intervjuer
 - Spørreskjemaer
 - Etnografi/observasjon

Aktivitet 3: Validering av kravspesifikasjoner

- Sjekk at kravene definerer det systemet som kunden faktisk vil ha
- Feil i krav koster mye
 - Å rette opp en kravfeil etter at systemet er utviklet og tatt i bruk koster svært mye mer enn å rette opp feilen mens kravene spesifiseres

Aktivitet 4: Håndtering av kravendringer

- Forretningsområde og tekniske omgivelser vil endre seg etter at systemet er tatt i bruk
- Brukere vil oppdage nye behov etter hvert som systemet tas i bruk
- Trenger oversikt over avhengigheter mellom kravene slik at man kan vurdere konsekvensene av å endre dem
- Trenger en formell (strukturert) prosess for å vurdere og evt. gjennomføre endringsforslag
 - Hvilken endring foreslås?
 - Hvem foreslår?
 - Hvem vurderer behovet for endringen, lager konsekvensanalyse, og beslutter om endringen skal implementeres?
 - Hvem skal involveres i implementeringen?
 - Hvem følger opp?

Kravspesifikasjon som grunnlag for testing

- For dårlig arbeid med kravspesifisering er én av årsakene til "IT-skandaler"
- Hvis kravene er dårlig spesifiserte, kan man ikke stole på at systemet er bra selv om systemet er testet mot kravspesifikasjonen

Oppsummering

- Det å innhente og analysere krav til et IT-system før det utvikles, er blant de viktigste aktivitetene i systemutvikling
- Det finnes en rekke metoder, teknikker og verktøy for å støtte en god kravhåndteringsprosess
- Selv om man jobber smidig som skal tilrettelegge for endringer underveis, må man ha god kravspesifisering for å sikre kvalitet, spare tid og redusere kostnader, og for å vurdere om man skal utvikle selv eller anskaffe hyllevare