

Systemutveckling och användbarhet

Användarcentrerad systemutveckling, gränssnitt och prototyper.

Referens till avsnitt i kursboken

- Dix kapitel 6
- Gulliksen, Göransson: Användarcentrerad systemdesign, kapitel: 4, 5, 6, 7, 8, 9 (Bredvidläsning)

Syfte med lektionen

- Att förstå grunderna till vad användarcentrerad systemdesign innebär, både i teorin och i praktiken. Vissa delar av det användarcentrerade projektet ska behandlas mer i detalj, t ex hur man använder sig av prototyper i utvecklingsarbetet.

Innehåll

- Olika systemutvecklingsmodeller.
- Vattenfallsmodellen kontra experimentell/iterativ utveckling.
- Användarcentrerad utveckling.
- Användarens roll i projektet.
- Kommerciella utvecklingsmodeller.
- Prototyputveckling.

Chaos om datorprojekt..

- 30% läggs ner innan de är klara
- 50% av projekten
 - kostar 190% av budget
 - till 40% av funktionaliteten
- 20% är klara i tid, inom budget
 - för *stora* projekt är siffran nere i under 10% (www.standishgroup.com)
- VARFÖR accepterar vi detta??
- Vad beror problemen på?

Vilka projekt lyckas?

- Förankring i organisationen om syfte och krav
- Användarmedverkan
- Fokus på användbarhet hela tiden
- Verksamheterna i balans
- Små projekt eller stegvis utveckling
- Utnyttjar beprövad teknik
- Genomtänkt och välplanerat införande

Användarcentrerad systemutveckling och prototyper

Denna bildserie sammanfattar viktigare aspekter på området.

- Olika systemutvecklingsmodeller.
- Vattenfallsmodellen kontra experimentell/iterativ utveckling.
- Användarcentrerad utveckling.
- Användarens roll i projektet.
- Kommersiella utvecklingsmodeller.
- Prototyputveckling.

Olika systemutvecklingsmodeller

I ett systemutvecklingsprojekt ska man i ett antal steg:

- ta fram underlag för design och konstruktion av ett nytt informationssystem,
 - konstruera systemet,
 - testa det,
 - införa – implementera det,
 - utvärdera och följa upp,
 - kontinuerligt underhålla och vidareutveckla.
- Det finns ett antal olika modeller och metoder för hur detta arbete organiseras och genomförs. Vi ska kort behandla ett antal aspekter på hur detta kan och bör göras för att man ska få ett bra slutresultat.

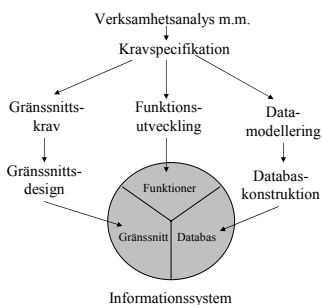
Vad består ett informationssystem av?

Förenklat kan man säga att ett informationssystem består av:

- Funktionalitet
- Databas – datalager
- Användargränssnitt

För att få en bra grund för det kommande utvecklingsarbetet genomför man initialt en verksamhets- och uppgiftsanlys. Därefter sker utvecklingen av de olika delarna, ett arbete som ofta sker med interaktioner mellan delarna.

Se figur!



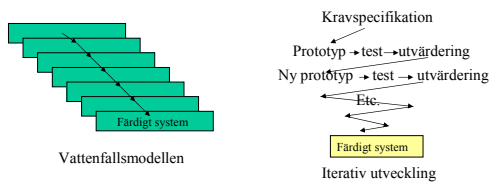
Organisation av utvecklingsarbetet

- För att få systemutvecklingsarbetet att fungera bra behöver man dela upp det i ett antal steg, arbeta enligt någon väl utvecklad och testad modell som ger stadga åt arbetet, identifiera de olika roller som ska samverka under processen m.m.
- Det finns flera olika sådana modeller som har provats tidigare och som är de som gäller i praktiken idag. Dessa har ofta gemensamma huvuddrag, men skiljer sig åt i vissa avseenden.
- En viktig aspekt som vi här vill trycka på är att bra modeller bör involvera användare på ett effektivt sätt samt stödja hur man behandlar frågor om användbarhet. Det är tyvärr vanligt att man i praktiken fokuserar på att få systemet att fungera rent tekniskt, och frågan om hur användbart systemet är får ofta låg prioritet.
- Användningen av prototyper underlättar arbetet!

Vattenfallsmodellen kontra experimentell/iterativ utveckling

Man brukar skilja mellan två huvudtyper av utvecklingsmodeller

- i den ena, "vattenfallsmodellen", sker de olika utvecklingsstegen i en sekvens, steg för steg tills systemet är klart,
- i den andra, experimentell eller iterativ utveckling, sker utvecklingen i ett antal cykler där man successivt förbättrar systemet tills utvärdering visar att man är nöjd. Kallas även spiralmodellen m.m.

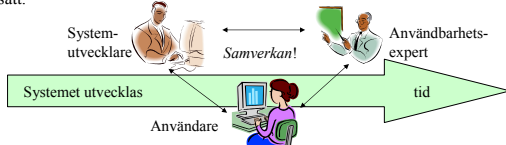


Olika utvecklingsmodeller

- Man kan säga att vattenfallsmodellen varit vanligast historiskt sätt.
- Vattenfallsmodellen ger ingen styrning genom utvärdering under tiden, och leder därför ofta inte till effektiva och användaranpassade lösningar
- Iterativa metoder med utnyttjande av prototyper är bättre i många avseenden.
- De iterativa metoderna bygger oftast på ett aktivt deltagande av användare, dvs *användarcentrerad utveckling*.
- Det brukar heta att det tar längre tid att arbeta iterativt och med användare med. Det behöver inte vara så! Det beror på hur duktig man är på att arbeta enligt en sådan modell.

Användarcentrerad utveckling.

- Användarcentrerad utveckling innebär att användare, dvs de som kan det arbete som det nya informationssystemet ska stödja, är aktivt med i projektet. De ska därvid svara för sin kompetens om sitt eget arbete.
- Systemutvecklarna ska vara experter på att utveckla det tekniska systemet, vilket ska bygga på de krav som användarna formulerar.
- Processen behöver ofta någon expert på användbarhetsfrågor med i projektet, vilka kan tolka användarnas krav, omvandla dessa till tekniska krav, genomföra tester och utvärderingar av prototyper osv.
- De tre rollerna måste förstå varandra och kunna samverka på ett effektivt sätt.



Användarens roll i projektet.

- Användarnas roll är viktig att precisera och ge dem utbildning i, eftersom de normalt inte utför detta arbete. De är ju experter på sitt eget arbete, och inte på att utveckla datorsystem.
- Användarna ska inte formulera sina krav i datatekniska termer utan i termer av hur man vill kunna utföra sitt arbete.
- Det kan vara svårt för användarna att kunna vara nog visionära, dvs kunna se möjligheterna i hur man skulle kunna arbeta med hjälp av det nya teknisktödet.
- Användarna ska vara med i varje steg av utvärderingen av de prototyper som tas fram. De måste därvid kunna se hur deras eget arbete kan komma att förändras efter det att systemet införts. Dvs prototyperna måste spegla deras normala arbetsituation.
- Användarna bör ges ett tydligt mandat i projektet. Man kan lätt bli överkörda av tekniker som "alltid vet bäst", som kan säga att "det går inte att göra så" osv. Man brukar säga att användarna blir "gisslan".

Kommersiella utvecklingsmodeller.

- Det finns många olika kommersiella utvecklingsmodell på marknaden, som avser att stödja systemutvecklingsprocessen. Dessa system är ofta mycket komplexa till sitt innehåll, består av ett stort antal olika steg, ger hjälp med planering, genomförande och dokumentation av de olika stegen osv.
- Ofta har modellerna ett tydligt teknikfokus, dvs de syftar till att effektivisera systemutvecklarnas arbete, inte att stödja arbetet med användare och användbarhet.
- Användningen av effektiva utvecklingsmodeller är nödvändigt i dagens praktiska systemutveckling. Annars skulle arbetet bli alltför svåröverskådligt och svårstyr. Men det är viktigt att se till att användbarhetsfrågor inte glöms bort pga att utvecklingsmodellerna inte stödjer detta nog bra!
- Ett exempel på en i dag vanlig modell är RUP, Rational Unified Process. Denna modell beskrivs i Gulliksens och Göranssons bok.

Prototyputveckling

- En grund för användarcentrerad utveckling är, som vi sett, användningen av *prototyper*.
- Prototyper är mer eller mindre fungerande system, som uppvisar ett utseende och beteende som illustrerar hur det tänkta nya informationssystemet kommer att se ut.
- Prototyper utvecklas från mycket enkla skisser kanske på papper, via mer kompletta och fungerande delsystem, till färdiga prototyper som visar hur det slutliga systemet kommer att fungera.
- Det finns idag bra tekniska system för att ta fram prototyper, men fortfarande brukar snabb framtagning av prototyper vara ett problem i projekten.
- I ett projekt måste man kunna ta fram nog många prototyper, dvs iterera nog många gånger, för att användarna ska få en chans att påverka systemutformningen. "Cykeltiden" för en iteration måste bli kort nog.

Prototyping...

Man skiljer på olika principer för prototyputveckling:

- *Requirements animation*. Enklare illustrationer utan att ha fungerande systemdelar.
- *Rapid prototyping*. Snabba prototyper som förkastas vartefter.
- *Incremental prototyping*. Stegvis utveckling av delfunktioner.
- *Evolutionary prototyping*. Systemet utvecklas från enkla versioner som successivt byggs på och till slut blir det färdiga systemet.

En prototyp kan antingen vara en ren illustration av det system som är under utveckling, eller en grund för det slutliga systemet.

- "Throw-away" prototyping
- Prototyping genom stegvis systemutveckling

Att fundera över

- Det visar sig att användarcentrerad systemutveckling är svår att genomföra i praktiken. Många projekt misslyckas. Vad tror du det beror på? Vad kan man göra för att förenkla processen, så att det fungerar bra i daglig systemutveckling?
- Man kan argumentera både för och emot tidig prototyping i ett projekt. Med tidig prototyputveckling kan man t ex låsa sig i tidiga lösningar och ha svårt att se nya möjligheter senare i projektet. Med sen prototyping kan användarna ha svårt att nog konkret förstå vad man håller på med, och när de förstår det är det redan för sent. Vad tror du att man bör göra? Använda prototyper tidigt och/eller sent i projektet? Vad kan man göra för att motverka de problem som kan uppstå?
