

Strategi og implementering:

# **Erfaringer med EPJ i Ribe Amt**

Esben Dalsgaard

IT-leder

Psykiatri- og Sundhedsområdet, Ribe Amt

[eda@ribeamt.dk](mailto:eda@ribeamt.dk)

# Status Ribe Amt

"Harmoni" er en 2. generations EPJ, der har været i drift siden 1999:

- Lagdelt (multi-tier) arkitektur
- Modulopbygget
- Baseret på Java2EE komponenter (Enterprise Java Beans)
- Modelbaseret

# Status Ribe Amt

- Version 1.0 blev taget i drift december 1999 på et enkelt sengeafsnit og dækkede al sygeplejedokumentation samt registrering af ordineret medicin.
- Version 3.3 vil inden udgangen af 2002 dække ca. 30% af sygehussengene i Ribe Amt (bl.a. fuld dækning på Grindsted og Brørup sygehuse). Indeholder bl.a. laboratoriesvar, skemaer, procedurehåndbøger, notifikation samt behandlingsplaner og Psyk-VIPS (psykiatrien).

# Status Ribe Amt

- Version 4.X vil inden udgangen af 2003 dække ca. 50% af sygehussengene i Ribe Amt og vil indeholde et tværfagligt, problemorienteret dokumentationsmodul (notatmodul, NORA\*), der er i stand til at udveksle oplysninger med andre EPJ'er via et udvekslingsformat baseret på Sundhedsstyrelsens Grundstruktur (G-EPJ).

Versionen vil også indeholde elektronisk booking (kommende udbud), men det er endnu ikke planlagt hvor mange afdelinger der vil have taget dette i brug i 2003.

\* Er p.t. under udvikling på Centralsygehuset Esbjerg Varde

# Nuværende udbredelsesplaner

År	senge	% dækket
2000	65	7
2001	100	11
2002	274	30
2003	448	50
2004	622	69
2005	796	88
2006	900	100

Skal revideres iht. nye mål (fuld udbredelse inden udgangen af 2005)

# Major barriers for implementing an Electronic Health Record

- 57% Lack of adequate funding or resources
- 40% Difficulty in justifying the investment in an EHR system
- 36% Inability to find a vendor or technical solution that addresses the needs of my organization
- 34% Difficulty in implementing an information solution in a rapidly changing environment
- 31% Inadequate or incomplete healthcare information standards, data sets, or code sets
- ...

# Major barriers ... 2002

- 59% Lack of adequate funding or resources
- 35% Lack of support by the medical staff
- 32% Inability to find an EHR solution or components at an affordable cost
- 31% Difficulty in creating a migration plan from paper to electronic health records
- 29% Difficulty in finding a satisfactory EHR solution that is not fragmented over several vendors or IT platforms
- 25% Inadequate or incomplete healthcare information standards, connectivity standards, data sets, or code sets
- ...

# Barrierer ...

- EPJ har forskelligartede formål, skal kunne alt og være løsningen på alle problemer
- Manglende lokal forankring af projekter
- Manglende enighed om løsninger
- Svært at lave løsninger til en verden præget af store forandringer
- Kan VIPS og G-EPJ sammensmeltes?
- Umodent marked
- Problemstillinger vedr. samarbejde mellem flere leverandører

# Forskellige formål

- At få bragt eksisterende læge- og sygeplejournals over på elektronisk form (samt andre faggruppers dokumentation)
- At samle informationer fra forskellige kilder: Eksisterende IT-systemer, medicoteknisk udstyr etc.
- At understøtte beskrive arbejdsprocesser i form af patientforløb, instrukser og standarder
- At give beslutningsstøtte i den kliniske proces
- At understøtte samarbejdet (CSCW) mellem forskellige faggrupper

# Forskellige formål

- "... I relation til visse patientforløb, hvor flere forskellige sundhedspersoner er involveret, kan det tillige være relevant at arbejde med fælles journaler ... " (vandrejournaler mellem sygehusvæsenet og praksissektoren)  
... (Strategi og analyse på sygesikringsområdet 2001)
- At understøtte patientforløb på tværs af sektorer
- At kunne udveksle oplysninger med IT-systemer i andre sektorer (privat og kommunal)
- At kunne udveksle oplysninger med andre EPJ'er

# Forskellige formål

- At indføre et paradigmeskift ift. den nuværende dokumentation i sundhedsvæsenet - fra kronologiske og monofaglige journaler til strukturerede, tværfaglige og procesorienterede journaler.  
... (National IT-strategi for sundhedsvæsenet 2003 - 2007)
- "... På sigt vil opsamling og strukturering af data til de kliniske databaser foregå via den elektroniske patientjournal..."  
... (National strategi for kvalitetsudvikling i sundhedsvæsenet 2002 – 2006)
- At stille aktivitetsoplysninger til rådighed for et kommende forløbsbaseret Landspatientregister (F-LPR).

# Forskellige formål

- " ... EPJ skal ændre patientens status; fra passiv modtager af behandling til aktiv deltager med indflydelse på egen helbredelse og behandling ... "  
... (Borgerpanelets slutdokument om EPJ, teknologirådet 2002)
- ... Etc., etc., etc.

# Forankring

- Løsninger laves til det enkelte sygehus, men koordineres (EPJ-basis og sygehusspecifikke dele)
- Udbredelsesprojekter indledes med udviklingsprojekter
- Projekter planlægges i fællesskab (kontrakt med leverandør udarbejdes i fællesskab)
- Lokale projektledere og projektdeltagere (men med støtte fra central funktion)
- Direkte kontakt mellem fremtidige brugere og systemudviklere (brugerkonsulenter kan være formidlere)

# Forankring: Projektorganisation

Styregruppe (på hvert sygehus):

- Sygehusledelse
- Sygehus IT-chef
- Projektleder (ofte med afdelingsledelsesansvar)
- Amtslig IT Seniorkonsulent
- Sundhedsfaglige personer (forskellige specialer, forskellige faggrupper)
- Projektansvarlig fra leverandør

# Forankring: Projektorganisation

Projektgruppe:

- Projektleder
- Sundhedsfaglige personer (forskellige specialer, forskellige faggrupper)
- Amtslig IT-brugerkonsulent
- Systemudvikler fra leverandør

Hertil inddrages referencegrupper på sygehusene og amtets testfunktion

# Forankring: Projektorganisation

I og med der er tale om en organisations- og systemudviklings-proces, er det ikke nok at undervise kommende brugere i brugen af et givet system.

For at klæde kommende projektledere og –deltagere bedre på ift. projektarbejde planlægger Ribe Amt afholdelse af kurser heri (formodentlig starten af 2003).

# Change is the only constant

“Everything in software changes. The requirements change. The design changes. The business changes. The technology changes. The team changes. The team members change. The problem isn’t change, per se, because change is going to happen; the problem rather, is the inability to cope with change when it comes.”

Kent Beck: “Extreme Programming Explained. Embrace Change”,  
Addison-Wesley, 2000

# Eksempel: Sygeplejejournal

Januar 2000: Elektronisk plejeplan (Kardex)

Januar 2001: VIPS-model

Januar 2003: Tværfaglig problemorienteret fællesjournal  
(NORA, G-EPJ)

# Datalogiske epoker og udfordringer

- 1945 - 55 Håndtering af (hardware-) simpelhed
- 1955 - 70 Håndtering af programmeringseffektivitet  
(stigende opgavekompleksitet)
- 1970 - 80 Håndtering af (software-) kompleksitet
- 1980 - 00 Håndtering af usikkerhed (opgavekompleksitet)
- 2000 - Håndtering af forandring

# Product-oriented view

“The referent system is considered as being essentially *static*, having two states: before and after the software system is introduced. The effects of the software system on its environment are considered to be predictable.”

Christiane Floyd: Outline of a Paradigm Change in Software Engineering  
in Bjercknes et al. Computers and Democracy, Avebury 1987

# Process-oriented view

“The referent system is considered as being *dynamic* and *evolving* in time; the software system will contribute to bringing about changes in the environment, which will in turn react on the reference system in an unprecidtable manner.”

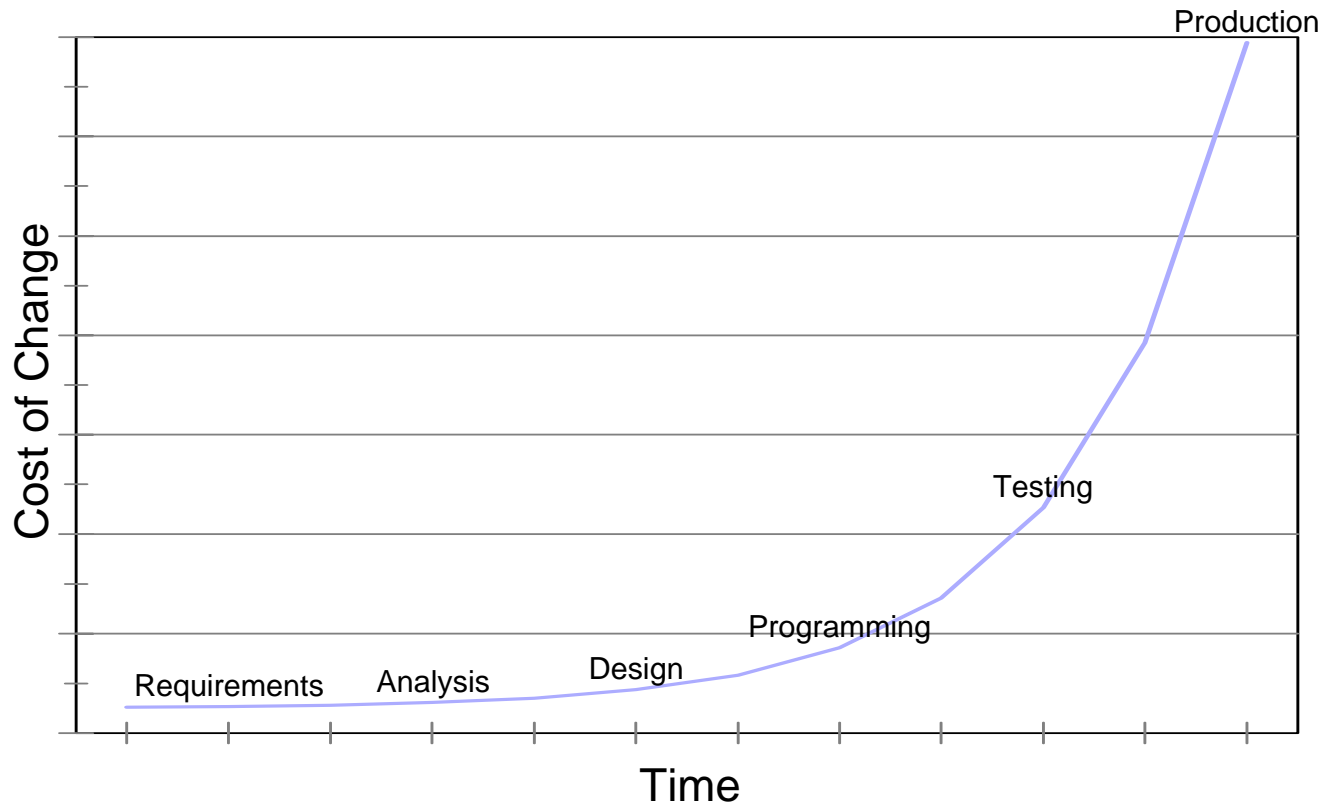
Christiane Floyd: Outline of a Paradigm Change in Software Engineering  
in Bjercknes et al. Computers and Democracy, Avebury 1987

# A paradigm change

"... the actual set of programs at any stage is considered as a *version*, subject to later revisions and embodying limited and possibly conflicting insight into different parts of the referent system."

Christiane Floyd: Outline of a Paradigm Change in Software Engineering  
in Bjercknes et al. Computers and Democracy, Avebury 1987

# Cost of Change



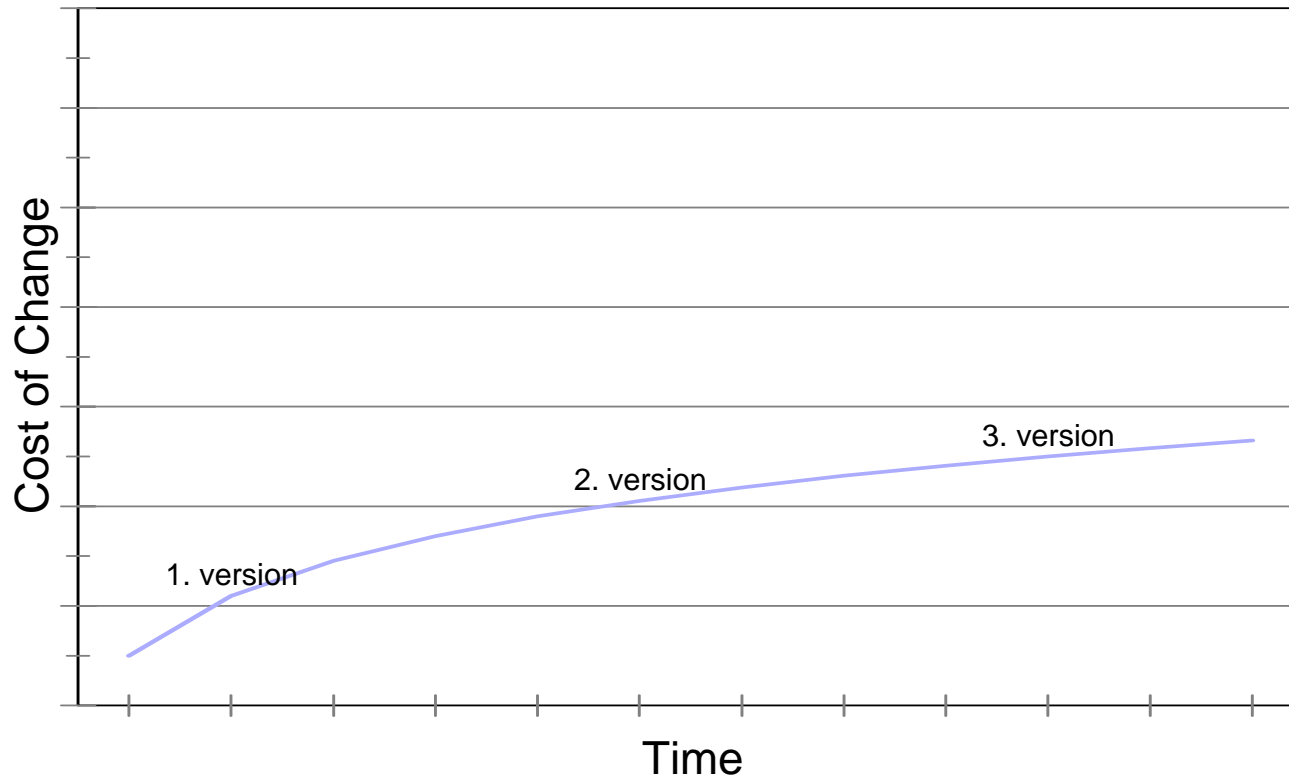
# Holder antagelserne?

Antagelse: Hvis omkostningerne ved at rette op på et problem i en kravspecifikation koster 1.000 kr., så koster det måske 100.000 kr. at rette op på samme problem, når først systemet er blevet udviklet og sat i drift.

Konklusion: Væsentlige problemstillinger skal identificeres og løses tidligt i et udviklingsprojekt.

... men hvad hvis antagelsen slet ikke holder mere ????

# Cost of Change



# Holder antagelserne?

Antagelse: Omkostningerne ved at ændre noget ændrer sig ikke væsentligt gennem udviklingsforløbet.

Konklusion: Udskyd væsentlige / svære problemstillinger til senere i projektet. Jo senere i udviklingsforløbet man træffer beslutninger, jo bedre grundlag træffes disse på.

# Fejltagelsen

Et succeskriterium for traditionelle metoder har været, at udvikle en løsning som dækker de behov man har beskrevet på bestillingstidspunktet frem for at levere en løsning, der dækker de behov man har på leveringstidspunktet.

Det er ikke et succeskriterium, at der bliver leveret det der er bestilt. Måske tværtimod - dette siger jo blot, at man ikke har lært noget i mellemtiden !

# Changes

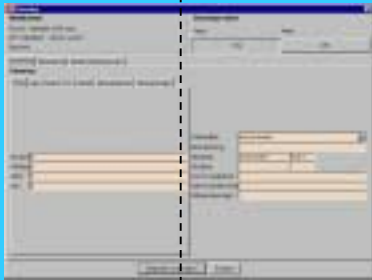
“They (the team) view changes to the requirements as *good* things, because they mean that the team has learned more about what it will take to satisfy the market.”

Robert C. Martin: “Agile Development: Principles, Patterns, and Process”,  
Prentice Hall, 2001

# Projektstyring

- Både visioner og mindre skridt er vigtige (tænk stort - start småt)
- Trinvis udvikling (tag en ting af gangen)
- Opdel i delprojekter på ikke over 6-8 mdr.
- Lad hvert delprojekt bestå af et antal leverancer (typisk 3)
- Læg funktionalitet i drift, så snart det kan give en nytteværdi (undlad ikke at lægge noget i drift, som giver en nytteværdi)
- Læg kun funktionalitet i drift der giver en nytteværdi

# Trinvis udvikling



**Grænseflade til  
andre systemer**

**Notat**

**Brugermodule**

**Fælleskomponenter**

**Domænemodel**

**Integrationslag**

**Data**

# Fordele ved trinvis udvikling

- Hurtig klinisk afprøvning af moduler og model
- Markedet modnes  
(der kommer efterhånden løsninger / moduler på markedet som kan anskaffes frem for at skulle udvikles,  
og man bliver efterhånden enige om modeller og standarder)
- Minimerer risiko ved projektstyring

# Ulemper ved trinvis udvikling

- Fare for knubskydning
- Funktionaliteten kan være for snæver til at give fuld nytteværdi (må eksempelvis suppleres med manuelle papirbaserede rutiner)
- Det kræver ressourcer med hyppige ændringer af systemet (men hvad koster det at benytte dårligt vedligeholdte systemer?)

# Konklusion

Det er vigtigere at have et modificerbart system end et system der på et givet tidspunkt har et givet udseende !

# Den datalogiske vinkel

Midler:

- Komponentbasing
- Robust arkitektur (byg på designerfaringer udtrykt ved "design patterns" og "application frameworks")
- Strategi om genbygning af systemer og løbende omstrukturering ("refactoring")
- Automatiser modultestning (opbyg testsuite sammen med systemet)

# Den datalogiske vinkel

- Traditionelle datalogiske dyder er vigtigere end tidligere: kravstyring (RM, requirements management), versionsstyring (CM, configuration management) og styring af kildekode (SCM, source code management)
- Vær opmærksom på nyere systemudviklingsmetoder der håndterer forandring (f.eks. agile methods)
- Udviklere skal forpligtes til at medvirke i nationalt standardiseringsarbejde (samt at overholde standarder)