



what makes you \*  
special?

IBM Softwaredag 2006



# Vindmøller og Software

Erik Kragh Dalskov  
Software Configuration Administrator





## Agenda

- **Vestas Wind Systems A/S og Vestas Technology R&D**
- **Implementeringsforløb**
- **Status for CM implementation**
- **Fremtiden**



## Vestas Wind Systems A/S

- **Etableret 1987**
- **Noteret på Københavns Fondsbørs 1998**
- **Fusioneret med NEG Micon 2004**
- **10618 medarbejdere (2005)**
- **Omsætning EUR 3.583 mio (2005)**
- **Markedsandel 28 % af installeret MW kapacitet i 2005 på verdensplan**
- **Vestas består i dag af 12 Business Units**
  - 4 produktions BU'er
  - 6 salgs BU'er
  - Vestas People & Culture
  - Vestas Technology R&D



## Vestas Technology R&D

- 550 medarbejdere
- 100 softwareudviklere
- 75 software udviklere bruger ClearCase
- 139 ClearCase projekter – 107 aktive og 32 obsolete
- 312 UCM komponenter – 291 aktive og 21 obsolete
- 200+ har adgang til fejlrapportering i ClearQuest
- 15 produktions ClearQuest databaser med et eller flere projekter i hver



## Implementeringsforløb

- **SPI (*Software Process Improvement*) projektet startet medio 2002 med nedsættelse af SEPG (*Software Engineering Process Group*) bestående af 1 fuldtids SPI projektleder samt 6 udviklere fra forskellige dele af softwareorganisationen**
- **De enkelte procesområder udvikles i et PIT (*Process Improvement Team*) bestående af 3-5 udviklere samt en koordinator fra SEPG.**
- **PIT arbejder ud fra et charter kommende fra SEPG**
- **Første fokusområder var Configuration Management, Review, Code Standard og Software Lifecycle Model**



## Implementeringsforløb – Configuration Management

- **Kompleksiteten af softwaren i vindmøller er vokset enormt**

1988	1994	1997	2001	2002	2007
385 kLOC	160 kLOC	340 kLOC	1.420 kLOC	1.310 kLOC	1.500 kLOC
1950 filer	665 filer	1050 filer	2650 filer	2450 filer	?
Release efter behov	Release efter behov	2-3 releases per år	2-3 releases per år	3-4 releases per år	?





## Implementeringsforløb – Configuration Management

- **Ønsker til CM:**
  - Sporbarhed mellem opgaver og kode
  - Versionering og opgavestyring sammenkoblet
  - Automatisk sammenhæng (baseline dependency) mellem komponenter ”på forskellige lag”
- **VSS (*Visual SourceSafe*) blev benyttet som versioneringsværktøj**
  - Problemer med stabilitet for store databaser
  - Ingen sammenhæng mellem komponenter





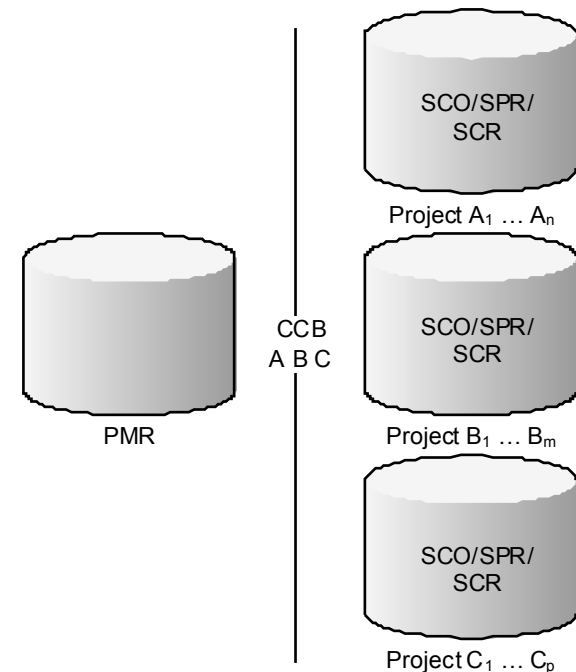
## Implementeringsforløb – Configuration Management

- **ClearCase(CC)/ClearQuest(CQ) valgt i maj 2003**
- **Første CQ schema version sommer 2003**
- **IBM konsulent involveret sommer 2003 – CQ schema får basal funktionalitet som stadig bruges**
- **Software til styring af møllerne lagt ind i CC i efteråret 2003 – 35 komponenter, 20 projekter - alt for komplekst**
- **Software til styring af møllerne lagt ind i CC ultimo 2003 igen – 2 projekter alt skrivbart – releasemanager rollen introduceret**
- **Efteråret 2004 – ny projektleder som bruger CQ aktivt til projektstyring – endelig udformning af overordnet design af CM system (CQ schema version 2).**
- **Forår 2005 – fuldtids CC/CQ administrator ansat**



## Nuværende ClearQuest implementation

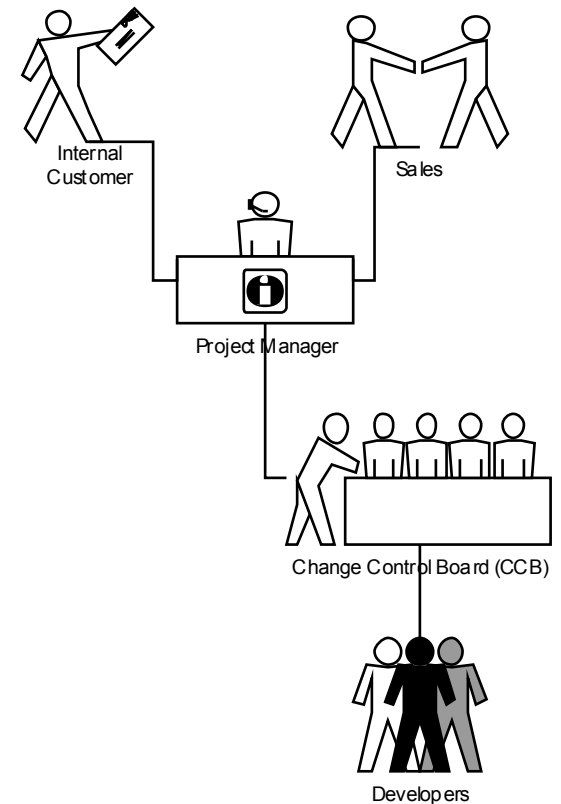
- Det nuværende ClearQuest implementation består af 2 niveauer
  1. Offentligt tilgængelig PMR (*Product Management Reports*) database
  2. Interne projektdatabaser.
- CCB (*Change Control Board*) tager stilling til alle PMR som kan
  - Blive rejected
  - Laves til en SCO (*Software Change Order*) – nyudviklingsopgave
  - Laves til en SPR (*Software Problem Record*) – fejlrapport
- Status på afledte opgaver opdateres på PMR så submitter altid kan se status uden at have adgang til de interne databaser





## Nuværende ClearQuest implementation

- **Change Management proces defineret ultimo 2004**
  - Interne kunder/Salg definerer PMR
  - Projektleder kvalificerer PMR
  - CCB godkender PMR og skaber SCO og SPR og prioriterer dem i forhold til eksisterende opgaver
  - Udviklere løser opgaven ved at checke filer ud og ind på SCR (*Software Change Record*), som tilknyttes en SCO eller SPR





## Nuværende ClearCase implementation

- **Unified Change Management (UCM) – aktivitetsbaseret udvikling**
- **Projekter bruger *release streams***
- **Andre ”lag” inkluderes *read-only* – giver lidt besværlig proces for ændring af disse**
- **Meget brug af ”*composite baselines*”**
- ***Check-out* og *check-in* foregår på SCR, som tilknyttes en SCO eller SPR**
- **Når en ny baseline skabes i ClearCase kan man fra de inkluderede SCR se hvilke SCO og SPR der indgår – *release note* genereret direkte fra koden!**



## Fremtiden

- **Projekterne skal selv styre brugeradgang til interne projektdatabaser**
- **Kravstyring i RequisitePro integreret i CM processen**
- **Mere stringent fejlrapporteringsproces**
- **SCRUM understøttelse i ClearQuest**



## Spørgsmål

