
5b. Konfigurationsstyring

Definition af konfigurationsstyring

Identifikation af konfigurations-komponenter

Konfigurations-kontrol og ændringsstyring

Konfigurations-status og -arkiv

Efter denne lektion (og
projektet) skal du:

- Kunne forklare hvad konfigurationsstyring er
- Kunne organisere konfigurations-håndteringen i et mindre projekt.



Konfigurations- styring

Af latin:

Com = Sammen,

Figurare = At forme

=> Styring af at forme
(IT) sammen

Kender du problemet ?

- En fejl du én gang har rettet dukker op igen !
- Du laver en forbedring til en god kunde. Straks efter udsendelse af en ny version klager kunden over at forbedringen er væk !
- En kunde ringer - men du kan ikke finde ud af hvilket program kunden har !
- En mystisk fejl får dig til at bruge dage på at granske kode - indtil du opdager, at kildeteksten du gransker ikke svarer til softwaren med fejlen i !
- Ingen kan finde ud af om en ændring, som det er besluttet at lave, er implementeret eller ej !

Oplæg til diskussion

Kender du nogen
af problemerne ?

Hvad er de værste eksempler
du kan komme i tanke om ?

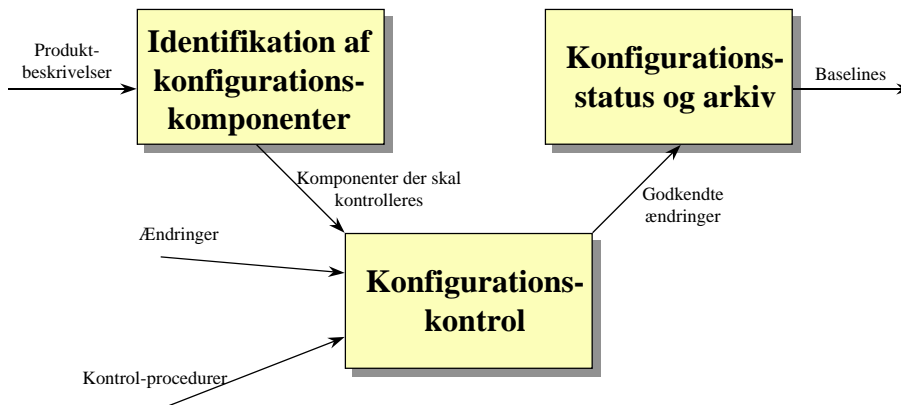
Tre kerneområder

**Identifikation af
konfigurations-
komponenter**

**Konfigurations-
status og arkiv**

**Konfigurations-
kontrol**

Relationer mellem kerneområder



Identifikation af komponenter 1

- Hvad består (software-) produktet af ?
- Hvad er den mindste del af produkt som I ændrer i ?
- Hvilken dokumentation er der knyttet til produkt, dels ved udvikling, dels ved levering til kunde ?
- Hvad er den mindste del af dokumentationen som I ændrer i ?

Identifikation af komponenter 2

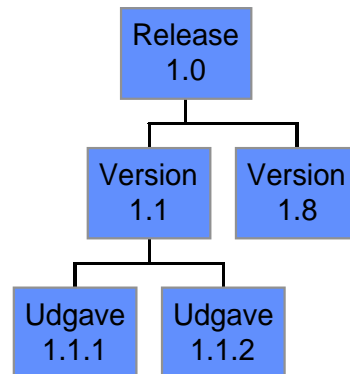
Hver komponent (dokument eller kode) skal have en identifikation, som:

- skal være entydig
- bør angive hvad komponenten er en del af
- bør angive hvornår komponenten blev skabt

Identifikation af komponenter 3

Komponenter indgår i et hierarki med f.eks.:

- Releases / Baselines
1.0, 2.0, 3.0 ...
- Versioner / Opdateringer
1.1, 1.2, 1.3, 1.4
- Udgaver / Smårettelser
1.1.1, 1.1.a, 1.11



Oplæg til diskussion i jeres projektgruppe

Hvilke komponenter vil I
have i projektet?
Hvor skal I gemme det?
Forskel mellem dokumenter
og programmer?

Typiske Baselines

En Baseline er et punkt man kan referere til,
som kan defineres efter hver fase i udviklin-
gen af et produkt, f.eks.:

- *Efter kravspecifikation => Funktionel Baseline*
- *Efter programdesign => Allokeret Baseline*
- *Efter moduldesign => Design Baseline*
- *Efter kodning og test => Produkt Baseline*
- *Efter integration => Operationel Baseline*

Funktionel Baseline

Ex. på indhold af en Funktionel Baseline (FB):

- *Forstudie Rapport*
- *Risikoanalyse rapport*
- *Produkt kravspecifikation*
- *Software kravspecifikation*
- *Accepttest specifikation*
- *Brugermanual - første udkast*

Produkt Baseline

Ex. på indhold af en Produkt Baseline (PB):

- *Udskrift af kode (kildetekst)*
- *Filer med Source kode*
- *Filer med objektkode*
- *Eksekverbar kode*
- *Modultest-rapporter*

Eksempel på dokumentation fra en virksomhed der laver isfabrikker

- **Anlægsbeskrivelse** - Anlæggets opbygning og funktion, oversigtsdiagrammer og figurer.
- **Systembeskrivelse** - Styresystem og ydre enheder, disponering af moduler.
- **Betjeningsvejledning** - Opstart, manuel drift, nedkøring, valg af forskellige funktioner
- **De ydre forbindelser** - Diagrammer og apparatlistes
- **Programdokumentation** - Flowdiagrammer (stigediagrammer), signallister, lagerdisponering

Oplæg til diskussion

Hvilke faser er der i jeres udviklingsproces ?

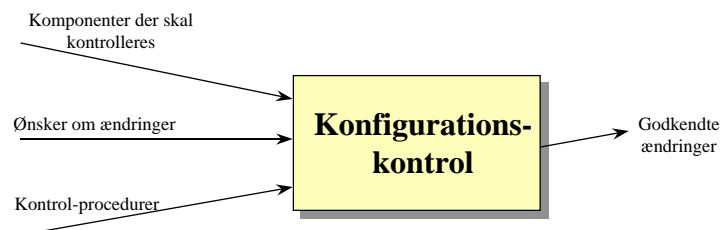
Hvilke Baselines kunne I med fordel have ?

Navngivning af komponent

PPP T IDID L S RR VV UU

<i>PPP</i>	<i>Projekt-initialer</i>
<i>T</i>	<i>Produkt-type</i>
<i>IDID</i>	<i>Komponent identifikationsnummer</i>
<i>L</i>	<i>Lagringsmedie (disk, bånd, papir)</i>
<i>S</i>	<i>Sted hvor komponent er lagret</i>
<i>RR</i>	<i>Release-nummer</i>
<i>VV</i>	<i>Versions-nummer</i>
<i>UU</i>	<i>Udgave-nummer</i>

Konfigurations-kontrol



Ændringer kommer fra to kilder:

- Fejl-rapporter
- Behov for forbedringer

Uanset kilden skal ændringsønsket vurderes, beslattes, implementeres, testes og godkendes.

Ansvar for ændringer

- Vigtigt at ansvaret for at VURDERE og BESLUTTE er helt klart placeret
- “Change Control Board” eller en gruppe med andet navn har ansvaret
- Ud af gruppens møder kommer Noter om godkendte ændringer (SCN = “Software Change Notice”)

Ændringsstyring

- Etabler et ændringsliste/log
- Udfyld en ændringsanmodning + konsekvens
- Godkend, afvis eller sæt som ønske
 - Projektgruppe
 - Styregruppe
- Opdater ændringsdokumentet
- Replanlæg og implementer ændringer i
 - Projektdokumentation og software



Oplæg til diskussion

Hvordan vil I registrere fejl og defekter f.eks. fra reviews?

Skal I have en formaliseret måde at afhjælpe fejl ?

Konfigurations-status og -arkiv



Software-arkivet

Kaldes også: Software-biblioteket

- Samlested for den officielle version af alle dokumenter og al kode
- Formel fremgangsmåde til at lægge nye eller rettede ting i arkivet (“tinglysning”), på basis af indkomne beslutninger fra konfigurations-kontrol
- Udlevering af arbejdskopier - og registrering af hvem der har/bruger hvad

Arkivets andre opgaver

- Klargøre det fulde produkt til levering: *Kun arkivet må klargøre ting til kunder*
- Genbrug: *Tilbyde oversigter. Registrere varianter*
- Dokumentations-center: *Opbevare alle breve, referater og noter. Stille dokumentations-historie til rådighed*

Oplæg til diskussion

Hvem arkiverer i projektet?
Hvad og hvor arkiveres?
Hvordan sikrer I at komponenter kan genbruges i næste semester?
Hvem gør produkt klar til levering til “kunde” (f.eks. lærer og censor i november)?