

# Programvaruutveckling för Stora Projekt

Christin Lindholm



## Föreläsning 4

- Granskningar
- Test, Konfigurationshantering
- Övrigt



## Vad ska ni göra?

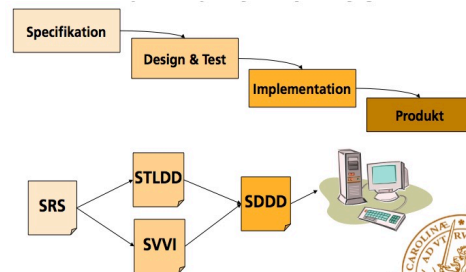
Tidrapporteringsystem Administration

Tidrapportering

Projektledning

Valfri funktionalitet

- Projektledare (PG)
- Systemansvariga (SG)
- Utvecklare (UG)
- Testare (TG)



## Risker



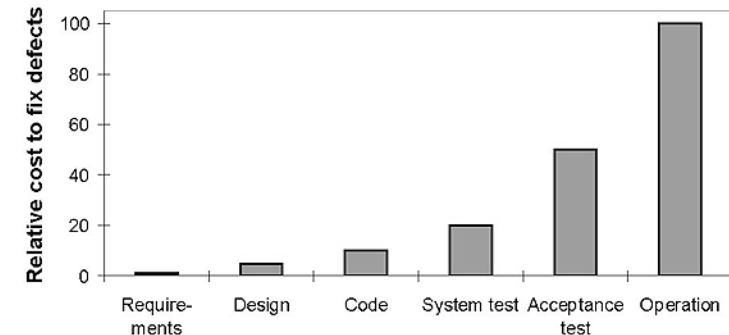
## Fas 1

1. Bilda projektgrupper
2. Tillsätta roller
3. Läsa in er på uppgiften och övrigt material
4. Arbeta efter utvecklingsmodellen
5. Skriva en kravspecifikation (SRS)
6. Skriva en projektplan (SDP)
7. Skriva en testspecifikation (SVVS)
8. Boka granskning (Granskningen veckan 6 eller 7)
9. Lämna in SDP, SRS och SVVS till granskning
10. Gå på granskningsmöte
11. Åtgärda kommentarer (ev omgranskning)
12. Baseline



5

## Felkostnader



[Alan Davies, 1992]



6

## Granskningar



- Grundläggande tanken enkel: läs granskningsobjektet systematiskt
- Dvs en manuell teknik med målen att:
  - Hitta fel
  - Sprida kunskap
  - Få beslutsunderlag
- Dessa mål kan vara olika mycket betonade i olika granskningsformer.



7

## Granskningar



- Viktig teknik för att hitta fel tidigt, istället för att "spara" dem till testfasen.
  - kan användas i alla steg i utvecklingen,
  - kan ligga till grund för formulering av ingångs- och utgångskriterier ("entry and exit criteria"),
  - kan utgöra värdefulla kontrollpunkter i utvecklingen,
  - ger oss en chans att kvantifiera kvaliteten.



8

## Om granskningar



- Olika undersökningar har visat på vinsten med granskningar.
- Omarbete efter granskning måste läggas in i tidplaner.
- Positiva bieffekter: spridning av kunskap, gruppkänsla och gemensamt ansvar.
- En granskning är ingen rättegång! Skilj på sak och person!



9

## Granskningars egenskaper

- Effektiv,
  - Ekonomisk,
  - Formell.
- Granskningar är en process: det finns faser och roller i samband med en granskning.



10

## Granskningsprocess

- Planering - rätt personer, lokal etc.
- Översikt - genomgång och distribution av material
- Förberedelser - stöds med checklistor från tidigare erfarenheter
- Granskningsmöte
  - Sammanställa felen - inte lösa problemen på mötet.
  - Granskningsprotokoll
    - olika typer av fel
    - granskningsbeslut (godkänd, godkänd efter ändringar, omgranskning)
- Omarbete - identifierade fel skall åtgärdas
- Uppföljning



11

## Resultat av granskningsmötet

Fyra resultat:

- Godkänt
- Godkänt med komplettering
- Omgranskning
- Mötet skjuts upp



12

## Granskningsroller

- Koordinator - ordförande, medlare
- Sekreterare - skriver protokoll
- Författare - svarar på frågor
  
- Granskare - har före mötet granskat (del av) dokument
  - Designer,
  - Programmerare,
  - Testare,
  - Kund,
  - Kvalitetsutvärderare,
  - Underhållspersonal.



13



## Granskningstekniker

- Checklistor
  
- Användningsfall

14



## Användning av data

- Kortsikt: Processtyrning (jmf projektstyrning)
  - exempel: estimering av kvarvarande fel
  
- Långsikt: Processförbättring



15



## Feltyper

- Feltyper för varje dokument (se PH:8):
  - SRS
  - SVVP & SVVR
  - STLDD & SDDD
  
- Generella feltyper om ej ovanstående passar
  
- Felgradering: A, B, C



Felklassificering är svårt - kom gärna med förslag på förbättringar.

16



## Exempel på ingångskriterier för granskning ("entry criteria")

- Tidigare dokument skall vara granskade, t.ex. design ska ej granskas förrän specifikationen är granskad.
- Eventuella standards har följts.
- Automatiska kontroller skall ha gjorts och vara godkända (t.ex. stavningskontroll)

17



## Erfarenheter

- Effektiviteten i granskningar kan mätas t.ex. m.h.a.:
  - medelantalet nedlagda timmar per hittat fel
  - antalet fel hittade per granskningsenhet (t.ex. per sida design)
- En del siffror:
  - Effektivitet: granskning är 2-10 gånger mer effektivt än test.
  - Granskning tar tid: 4-15% av ett projekts tid
  - Kan vi motivera detta?
    - Granskningar förbättrar kvaliteten.
    - Granskningar förbättrar produktiviteten.
- Exempel: fel i drift \$10.000, vilket kan bekosta många granskningstimmar.
- Kan granskningar ersätta test?

18



## Granskningar i projektet

- Informella granskningar - kund ej representerad, görs före varje formell granskning.
- Formella granskningar:
  - Software Specification Review
    - SDP, SRS, SVVS
  - Preliminary Design Review
    - STLDD, SVVI
  - Product Review
    - SVVR, SSD, PFR
    - Acceptanstest
- Ju bättre informell granskning desto bättre formell granskning

19



## Granskare

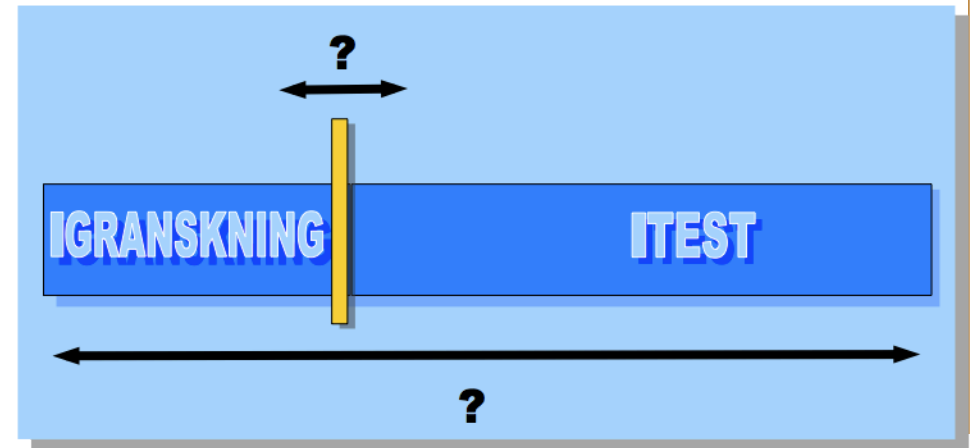
- Grupp 1 – Emelie Engström
- Grupp 2 – Anders Bruce

20



# Informell granskning

- Bestäm hur er informella gransknings process ska se ut
  - Vad ska granskas
  - När ska det granskas
  - Av vem (specifika namn)
  - Vem är ansvarig, sammankallande
  - Granska enskilt ->diskutera sedan problemen
  - Vem åtgärdar? Vem kollar?



22

# Success factors

- Ägandeskap över processen
- Stöd av ledningen
- Träning



23

# Nackdel

- Kostar mycket initialt
  - Slarv - slöseri med tid och pengar

Dock blir ett dokumentets kvalitet alltid lite bättre även vid dåligt genomförd granskning



24

## Begrepp

- **Validering**
  - Är det rätt system?
  - Uppfyller vi kundens behov och förväntningar.
- **Verifiering**
  - Är systemet rätt?
  - Har vi gjort det vi sa vi skulle göra?
- **Granskning**
  - Uppfyller ett objekt sin specifikation
- **Testning**
  - Uppfyller ett exekverbart objekt sin specifikation.

25



## Test

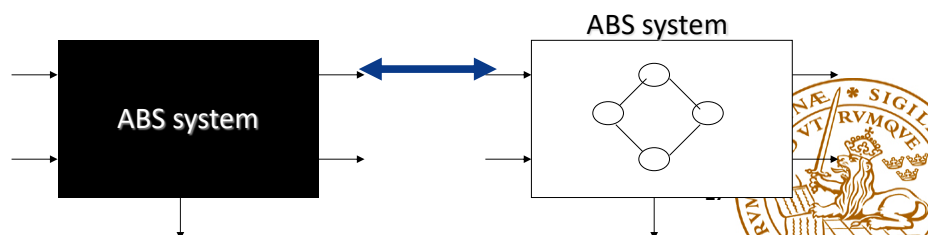
- **Målet för test är**
  - att hitta fel i programvaran
  - att åstadkomma förtroende för produkten
  - inte att bevisa frånvaron av fel.
- **En stor programvaruprodukt bör testas av**
  - utvecklaren och
  - en oberoende testgrupp;Testgruppen skall vara oberoende från implementationen och utgå från kravspecifikationen.

26



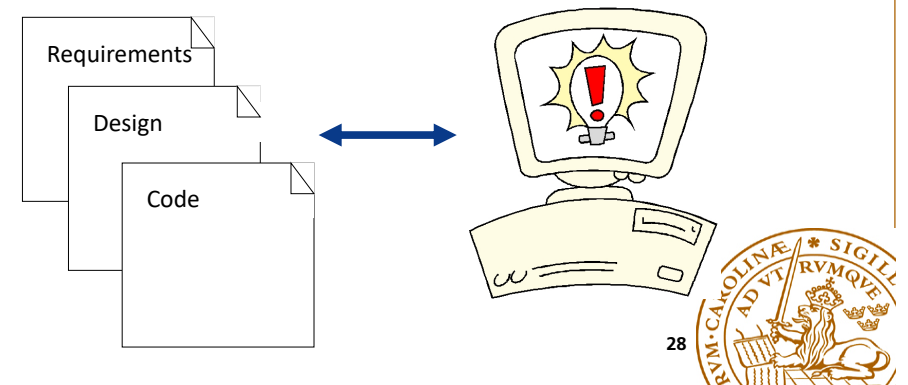
## Testmetoder (PH:7)

- **Två utgångspunkter:**
  - Whitebox -> utgår från interna strukturen
  - Blackbox -> utgår från externt observerbar funktionalitet



## Två angreppssätt

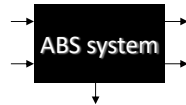
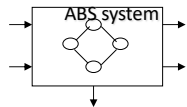
- Statiskt -> manuell granskning
- Dynamiskt-> exekvering av testfall



28

# Olika testtekniker

- **White-box**
  - Coverage
    - Decision
    - Statement
    - Branch
  - Ad Hoc
- **Black-box**
  - Equivalence Partitioning
  - Boundary Value
  - Random Testing
  - Error Guessing
  - Ad Hoc

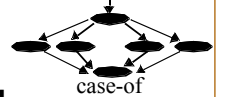
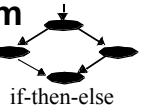
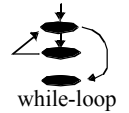
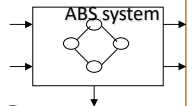


29



# Decision/Branch coverage

- **McCabe cyclomatic**
  - The number of independent paths needed
  - to cover all paths at least once in a program
    - Count number of conditional expressions. If compound conditional expressions, add one per compound item.
    - Visualize by drawing a flow graph
    - Alternative way:  $CC = \#(\text{edges}) - \#(\text{nodes}) + 1$

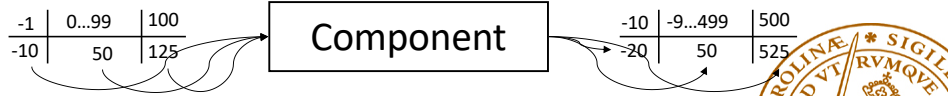
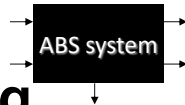


30



# Equivalence partitioning

- **Systematic way to choose test cases and data values**
  - Input or output data with common properties
  - Choose one test case per partition

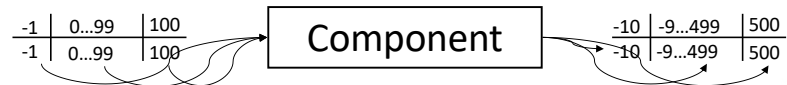
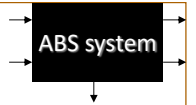


31



# Boundary value analysis

- **A systematic way to choose test cases and data values**
  - Input or output data with common properties
  - Choose one test case on each side of the boundary



32





## Olika testnivåer

- Enhetstest
- Funktionstest
- Integrationstest
- Systemtest
- Acceptanstest
- Installationstest

33



## Regressionstest

- Omtestning av tidigare testfall i samband med:
  - felrättning
  - ändringar
  - tillägg till ett existerande system
- Regressionstest är tämligen enkelt:
  - Man har redan testfallen
  - Man har ett tidigare utfall, dvs stämmer det fortfarande?

Hur mycket behöver vi regressionstesta?

34



## Testprocess

1. Identifiera alla testfall -> testspecifikationer (SVVS).
2. Beskriv varje testfall, steg för steg i naturligt språk -> testinstruktion (SVVI).
3. Testa och notera fel och rapportera
4. Regressionstesta och fortsatt testa med nya testfall
5. När ni är nöjda: kontakta kunden för acceptanstest

Varför behövs testspecifikationer? Testinstruktioner?

35



## Att tänka på

- Testspecifikation
  - Kort och kärnfull formulering
  - Unik numrering
  - Täcka alla (funktionella) krav
  - Referens till krav
- Testinstruktion
  - Systemläge vid start
  - Systemläge vid slut
  - Förväntat resultat
  - "self contained"

36



## Konfigurationshantering (Configuration Management - CM)

- CM = kontrollerat sätt att hantera utveckling och förändring av sammansatta system under hela livscykel
- Varför behövs CM?
  - Parallellisera arbetet
  - Veta i varje läge av utvecklingen hur olika delar beror av varandra och vilka versioner som gäller
  - Uppdateringsstrategi – hur inför vi ändringar/rättningar?

37



## Olika aktiviteter i CM

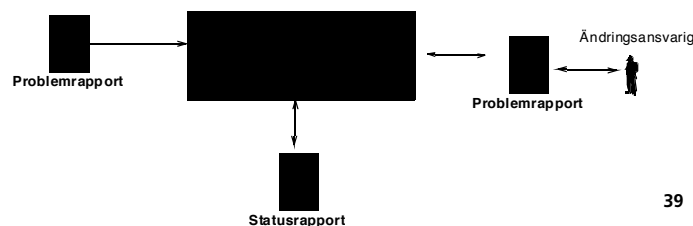
- Identifiering av konfigurationsenheter
- Benämning av versioner
- Ändringshantering
- Statusrapportering
- Systembyggande – sätta samman delar till en helhet

38



## Change Control Board (CCB) Förändringskontrollgruppen (FKG)

- Fattar beslut om ändringar
- Statusrapport anger historik för varje konfigurationsenhet
- Problemrapport – en slags stafettpinne som dokumenterar besluten och åtgärderna för en specifik ändring/rättning



39



## Identifikation av konfigurationsenheter

- Att peka ut och namnge de olika enheter som ska ändringshanteras
- Regler för namngivning: PUSP124819
  - Pusp=var producerat, 12=år 2012, 4= producerat av projektgrupp (typ av dokument, 0=kursmaterial), 8=projektgrupp, 19=löpnr
- Dokument, delar av dokument, systemdelar i arkitekturen, kod
- Vid inlämning till formell granskning 1 skall en lista på konfigurationsenheter skickas med
- Konfigurationsenhetslistan är en konfigurationsenhet!!!
- För varje konfigurationsenhet ska en statusrapport skapas

40



# Benämning av versioner

## – Version x.y

- Vid uppdateringar ökas y med 1
- Vid upprättande av baselines ökas x med 1
  
- Första versionen: 0.1
- ...0.9, 0.10, 0.11, ...
- Första baselinen: 1.0
  
- Baseline till kund kallas "Release" (utgåva)

41



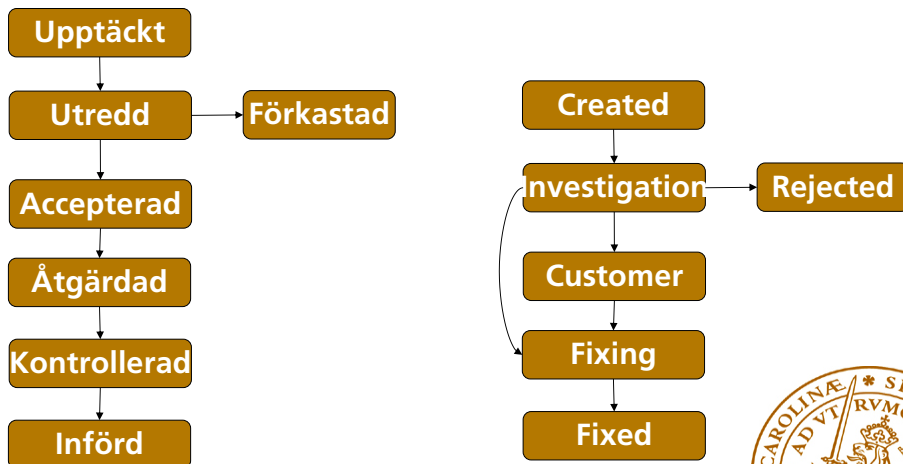
# Statusrapportering

STATUSRAPPORT						
Dokument: <i>SRS</i>			Dokumentansvarig: <i>Kent Larsson</i>			
Förändringskontrollansvarig: <i>Lars Björklund</i>						
DEL A: Dokumentstatus						
Pr nr.	Åtgärdas ja/nej	Ansvarig för åtgärd	Deadline datum	Version Nr. efter ändring	Godkänd datum	Kommentar
				0.1		Första version skapad
				0.2		till. Inform.granskning
				0.3		Till formell granskning
PR4	ja	MJ	990916	1.0	990917	Baseline
PR8	ja	MJ	990918	1.1	990918	Kravändring
PR9	nej	PP	990918			Nytt krav

42



# Livscykeln för ett problem



43



# Problemrapport

- Del A: Ursprung (Origin)
  - Var blev det problem? Vad är problemet?
- Del B: Utredning & Beslut (Investigation)
  - Ska problemet åtgärdas? Hur ska det åtgärdas? Vad påverkas?
  - Hur lång tid uppskattas åtgärden ta?
- Del C: Åtgärd (Action)
  - Åtgärdsansvarig?
  - Hur lång tid tog åtgärden att göra?
- Del D: Uppföljning & avslut (Follow up and completion)
  - Vem kontrollerade och godkände åtgärden?
  - När blev det färdigt?

44



## Slutsats



- Storskalig utveckling kräver processer, organisation och verktyg som kan hantera ökad komplexitet
- Det flesta av er kommer att arbeta i
  - Stora företag
  - Små och medelstora företag som levererar programvara eller system till stora företag
- Era ingenjörsfärdigheter är beroende av
  - Hur ni kombinerar teknologi och ekonomi
  - Arbetar i team och med stora organisationer

45



## Att göra!

- Förberedelseuppgift workshop 1 (Ing 2a)
- Workshop 1 (Ing 2a)
- Boka tid för formell granskning
- Göra laborationer
- Informell granskning. Åtgärda fel ni hittar
- Obligatorisk närvaro vid formella granskningar
- PG: Se till att allt är i ordning (inkl. mötesprotokoll etc.)
- PG: Se till att statusrapporter finns för SRS SVVS, SDP, Konfig.lista

46



## Kamratbedömning

- 2 ggr
  - Granskning 2
  - Acceptansmöte



47



## Individuella uppgiften

- Personligt brev till CV
  - Stöd i skrivandet av den individuella slutrapporten!
- Föreläsning den 5 feb
- Inlämning Studieverkstaden den 1/4 kl. 08.00 via mail
- Seminarium den 6/4 och 7/4
- Slutinlämning via mail 9/4 kl. 24.00

48



Obligatorisk närvaro!

# Workshop

**Workshop 1 (Ing 2a) – 5/2 kl. 08-12 i sal E230**

- Projekt
- Individuella uppgiften

**Workshop 2 (Ing 2b) – 8/4 kl. 08-12 i sal E210**

- Fokus på arbetslivet och åk 3

