

TDDI02

Programmeringsprojekt. Föreläsning 2

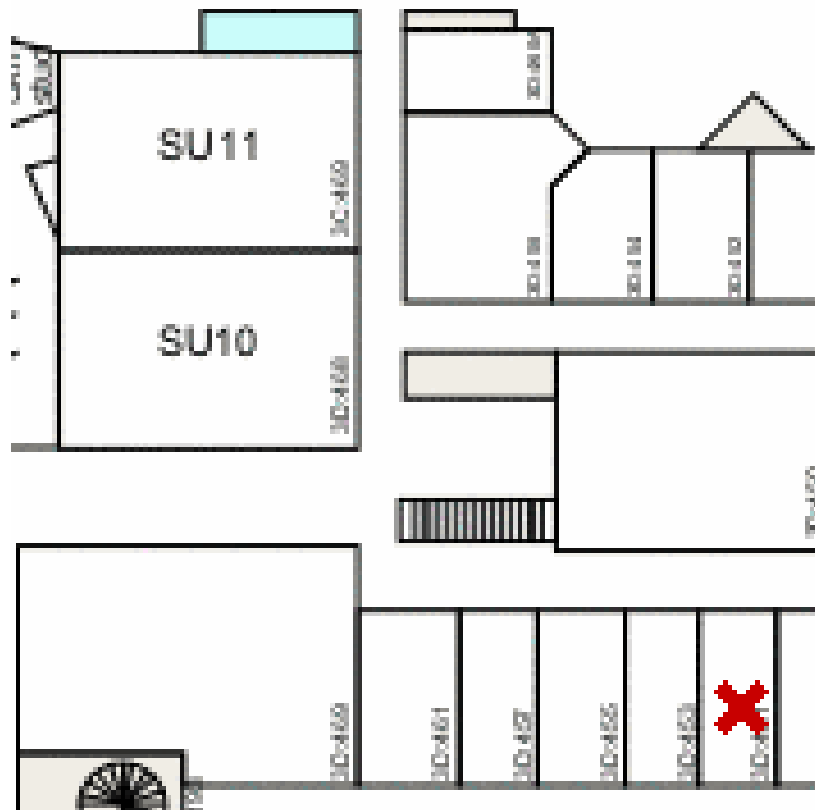
Jonas Lindgren, Institutionen för Datavetenskap, LiU

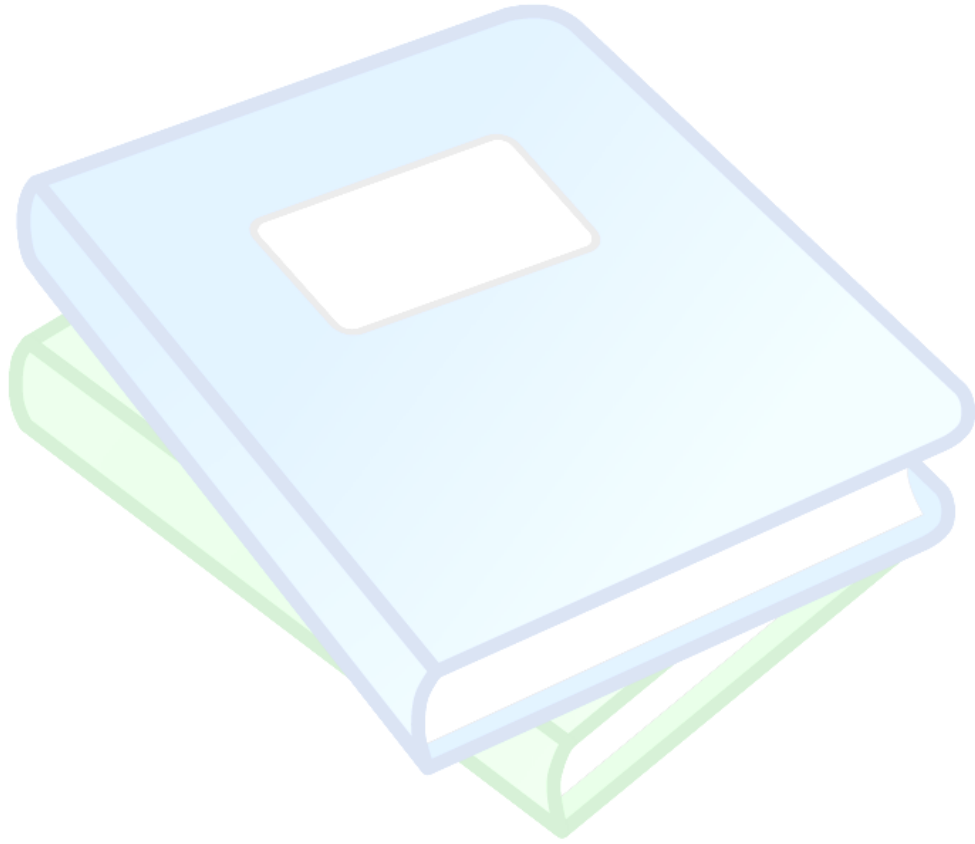
På denna föreläsning:

- Anatomien hos en projektplan
- Vad är klok design?

Tidsbokning

- Bokningslistor på Jonas Lindgrens dörr
- Ge minst en hel arbetsdags ledtid





Projektarbete kräver..

- Fördelning av arbete mellan grupper och individer, eventuellt med hänsyn till kompetenser
- Flitig kommunikation mellan grupper/individer
- Beroende mellan grupper/individens alster
- En tidsplan
- Kontrollpunkter för att avgöra om planen måste ändras
- Dokumentation, för att
 - Beskriva mål och förlopp för eventuella ersättare
 - Samla erfarenheter
 - Förbättra processen

Way-of-Working

The tailored set of practices and tools used by a team to guide and support their work.

- Good way of working is agreed by the team
- Good way of working reduces risks and technical debts
- Good way of working is effective and removes duplicate work and wastes
- Good way of working improves itself

Projektarbete kräver.. (cont.)

En projektplan skall (åtminstone)..

- Vara nedskriven i förväg
- Beskriva:
 - Vad som ska uträttas
 - Med vilken metod
 - *Ingen design!*
- Vara välskriven, strukturerad, kortfattad och begriplig
- Ta hänsyn till tänkbara ”olyckshändelser”

En projektplan kan innehålla..

- Översikt
Intro till jobbet, kunden, gruppens egen organisation
- Fasplan
Vilka utvecklingsfaser, vilka produkter, vilka datum?
- Organisationsplan
Vilka team, vems ansvar?
- Testplan
Vem? Hur? Verktyg?
- Förändringsplan
Vad händer om förutsättningar ändras?
- Dokumentationsplan
Vilka, när, till vem? Vem godkänner?

En projektplan kan inn.. (cont.)

- Utbildningsplan
Internt, externt. Vem, när, resurser?
- Plan för rapportering och "reviews"
Vad, till vem, när?
- Installationsplan
Vilken procedur krävs för att få igång?
- Plan för kvalitetssäkring
Standarder som ska användas?
- Varuplan
Vad ska levereras, när? Del-leveranser?
- Resursplan
Persontid, datortid. Summering av milstolpar!

Exempel på milstolpar

Fas	Milstolpe	Planerat datum	
Förstudiefasen	Förstudiedokument klart	1994-02-05	
	Definitionsfasen	Kravspecifikationsdokument klart	1994-02-25
		Kontrakt skrivet	1994-02-25
		Acceptansvillkor framställda	1994-02-25
		Projektplan klar	1994-03-04
Designfasen	Designdokument klart	1994-04-15	
	Programmerarhandbok klar	1994-04-13	
	Systemtestfall framställda	1994-04-05	
	Integrationstestfall framställda	1994-04-08	
	Modultestinstruktioner framställda	1994-04-15	
Programmeringsfasen	Teknisk dokumentation klar	1994-05-06	
	Användarhandledning klar	1994-05-06	
Testfasen	Modultest klara	1994-04-22	
	Integrationstest klar	1994-05-03	
	Systemtestning klar	1994-05-06	
Avslutningsfasen	Efterstudiedokument klart	1994-05-20	
	Acceptansöverenskommelse klar	1994-05-10	
	Slutrapport från kvalitetsarbete klart	1994-05-20	
	Återkoppling klar	1994-05-20	

Anmärkning: För täta, för "små". Borde nog innehålla mer av "godkänt/accepterat".

Work

Activity involving mental or physical effort done in order to achieve a result.

-
- Healthy Work is sizeable, estimate-able and track-able
 - Healthy Work breakdown reduces dependencies between work items
 - Healthy Work management keeps risks, work and re-work under control

Planering: GANTT-diagram

WEEKS: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23

WBS 1 Summary Element 1



WBS 1.1 Activity A



START-TO-START

WBS 1.2 Activity B



FINISH-TO-START

WBS 1.3 Activity C



FINISH-TO-FINISH

WBS 1.4 Activity D



WBS 2 Summary Element 2



WBS 2.1 Activity E



WBS 2.2 Activity F

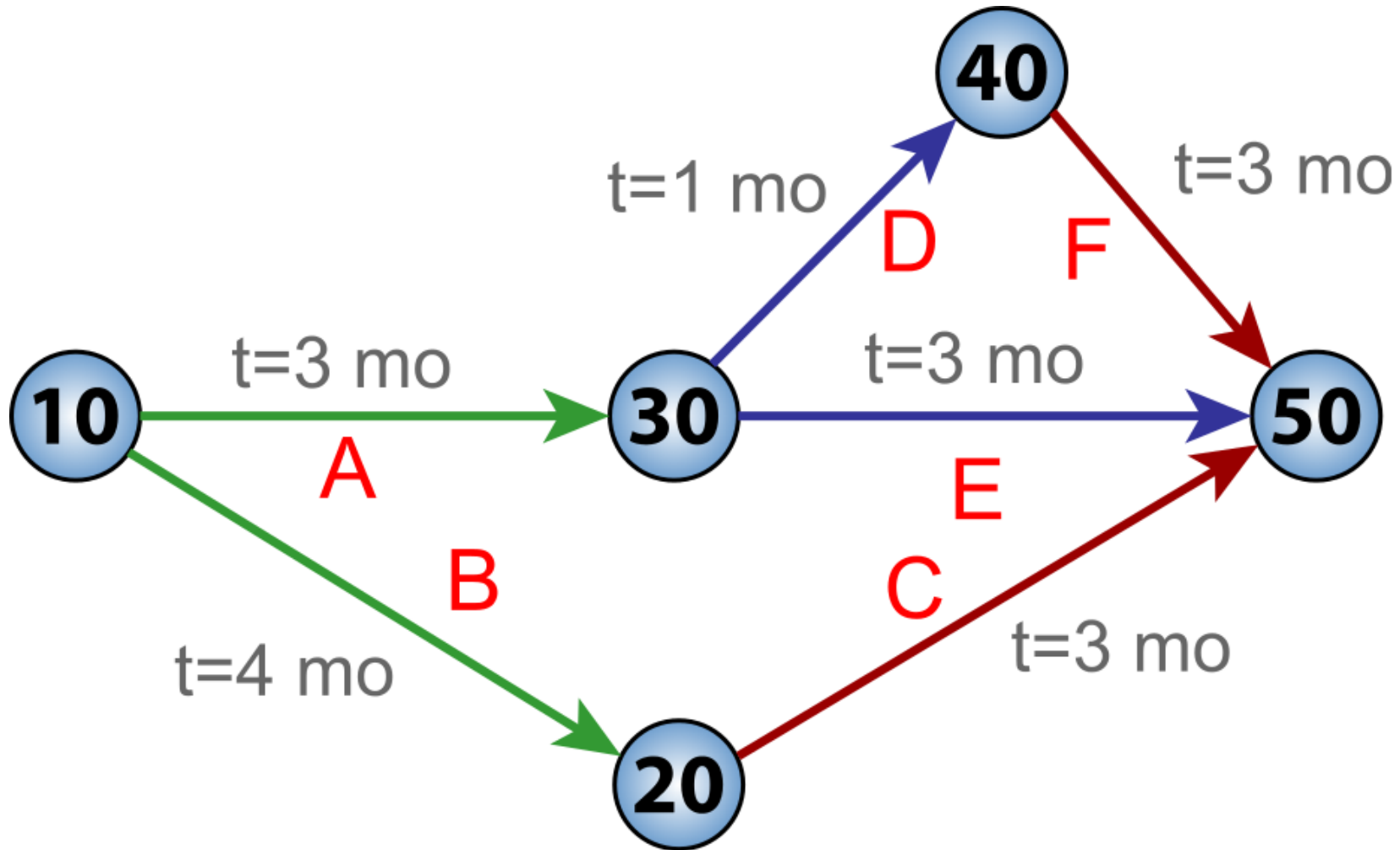


WBS 2.3 Activity G

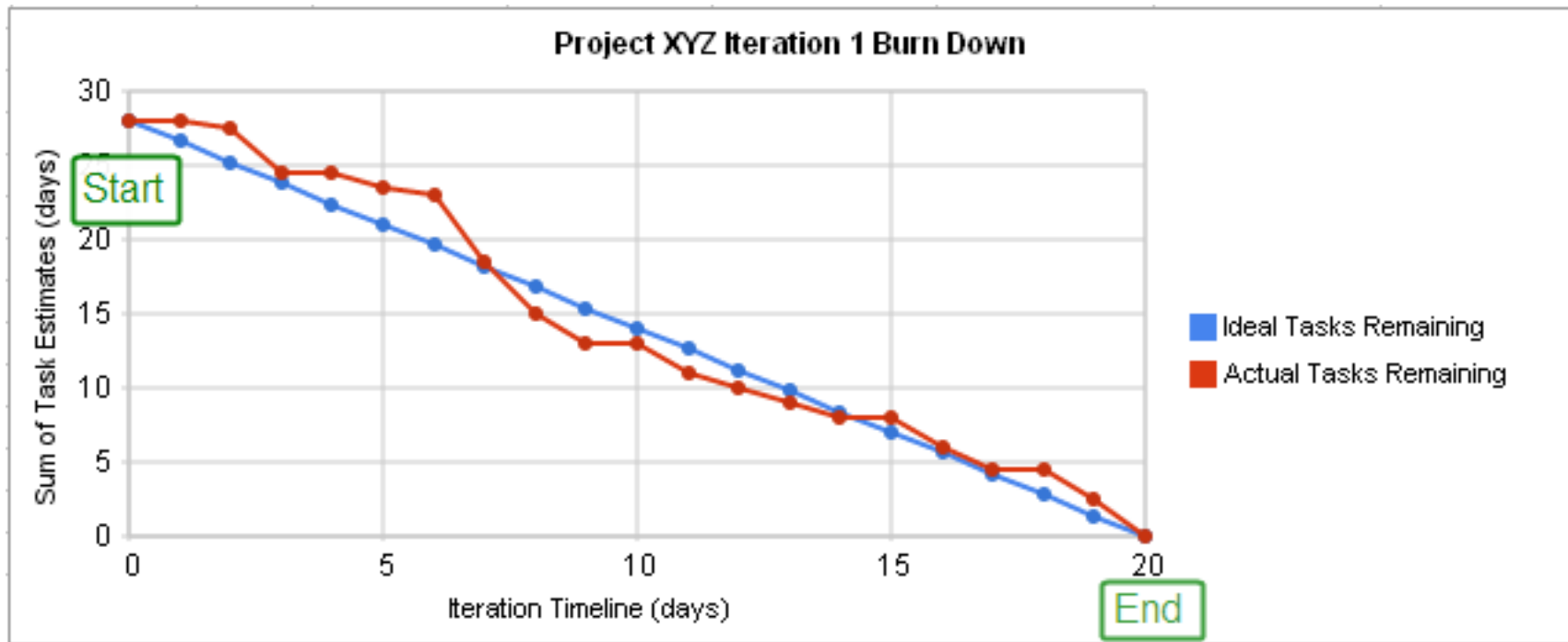


TODAY

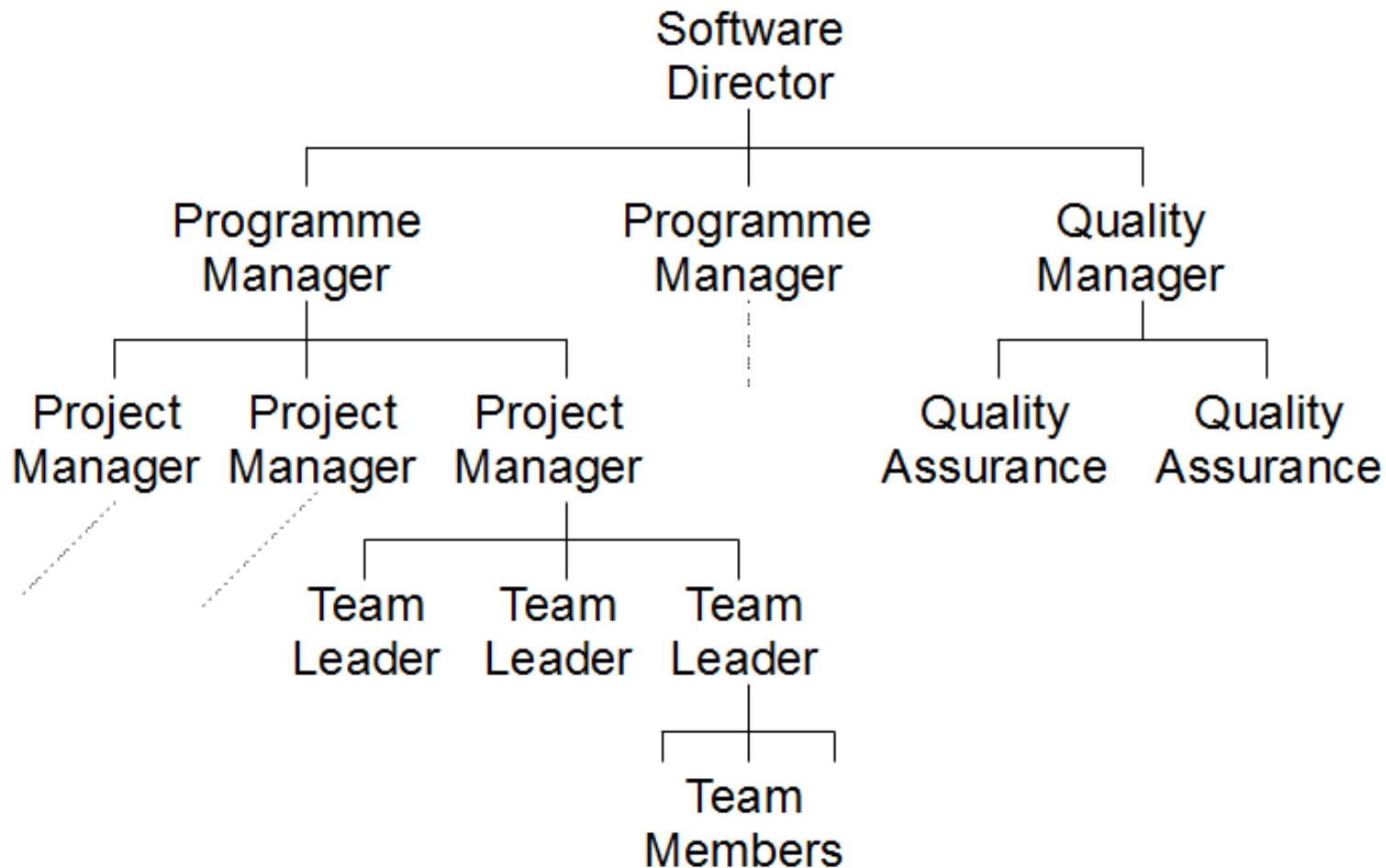
Planering: PERT-diagram



Planering: Burn down chart

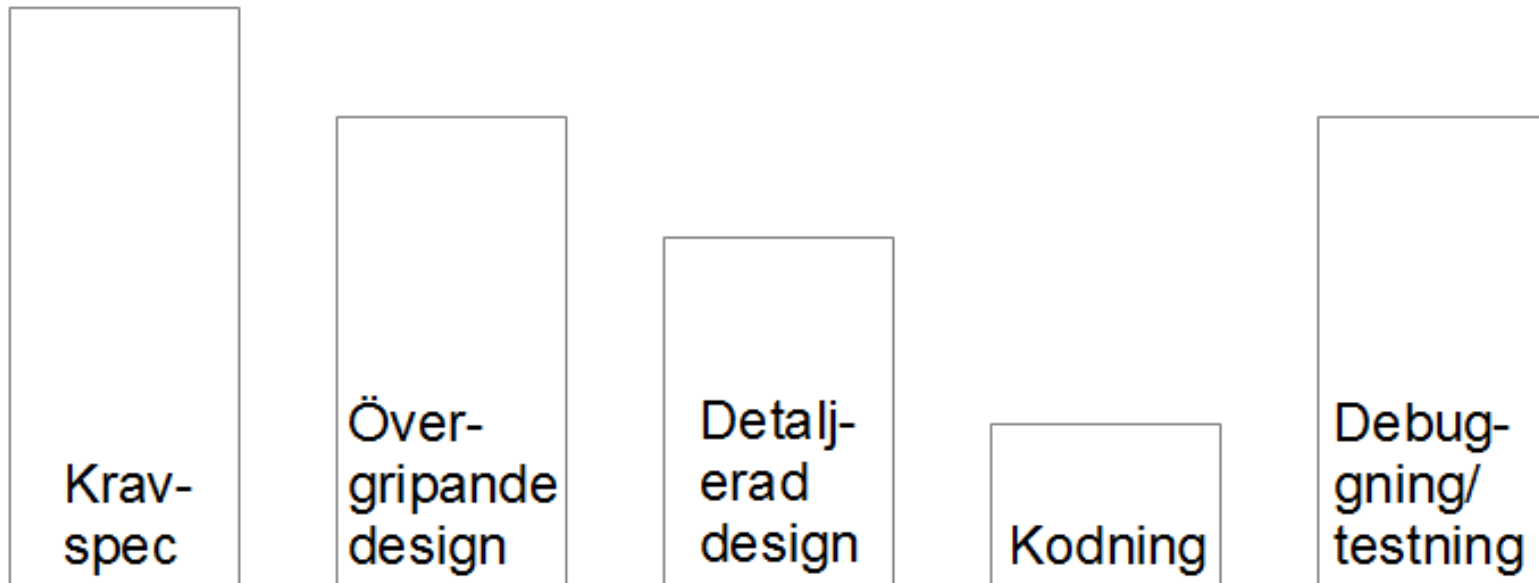


Exempel på organisation



Kompetenskrav?

Graden av "skicklighet" som kan krävas för olika faser eller arbetsmoment vid programvaruutveckling.



Rubriker i "verklig" projektpl.

1. Revisionshistoria
 - 1.1 Ändringslogg
 - 1.2 Relaterade dokument
2. Förutsättningar och bakgrund
 - 2.1 Syfte
 - 2.2 Bakgrund
3. Mål
 - 3.1 Affärsmål
 - 3.2 Systemmål
 - 3.3 Kvalitetsmål
4. Omfattning och resultat
 - 4.1 Projektets uppgift
 - 4.2 Avgränsningar
 - 4.3 Förväntade resultat
5. Kopplingar till andra projekt

Rubriker i "verklig" pro.. (cont.)

6. Projektorganisation

6.1 Styrgrupp

6.2 Projektorganisation

6.3 Ansvar

6.4 Ansvarsmässig avgränsning

6.5 Projektmöten

6.6 Samverkan och rapportering

6.7 Resursplan

7. Arbetsmetodik

7.1 Arbetsmetod

7.2 Verifiering

7.3 Validering

7.4 Upphandling/köp

7.5 Kommunikationsplan

7.6 Testplan

7.7 Kvalitetsplan

Rubriker i ”verklig” pro.. (cont.)

- 8. Tidplan och milstolpar
 - 8.1 Milstolpar
 - 8.2 Tidplan
 - 8.3 Leverabler till kund
 - 8.4 Utrullning
 - 8.5 Kriterier för överlämning
- 9. Kostnader
 - 9.1 Utvecklingsmiljö
 - 9.2 Testmiljö
 - 9.3 Produktionsmiljö
- 10. Risker
 - 10.1 Beroenden
 - 10.2 Stabilitet
 - 10.3 Prestanda
 - 10.4 Andra identifierade risker
- 11. Projektavslut
 - 11.1 Överlämning till drift och förvaltning
 - 11.2 Utvärdering
- 12. Ändringshantering

Design – Viktiga nyckelord

- Modularisering
 - Små, överblickbara bitar
- Modulkvalitet
 - ”Bra” egenskaper som kännetecknar en modul:
 - Information hiding
 - Göm viss detaljerad ”kunskap”
 - Abstraktioner
 - Access via gränssnitt
 - Språkoberoende
 - ”Högre” nivå, mer oberoende implementation
- Metodik
 - Hur man urskiljer moduler

Modul

Viktigast: Överblickbar, förståelig för en person eller ett team.

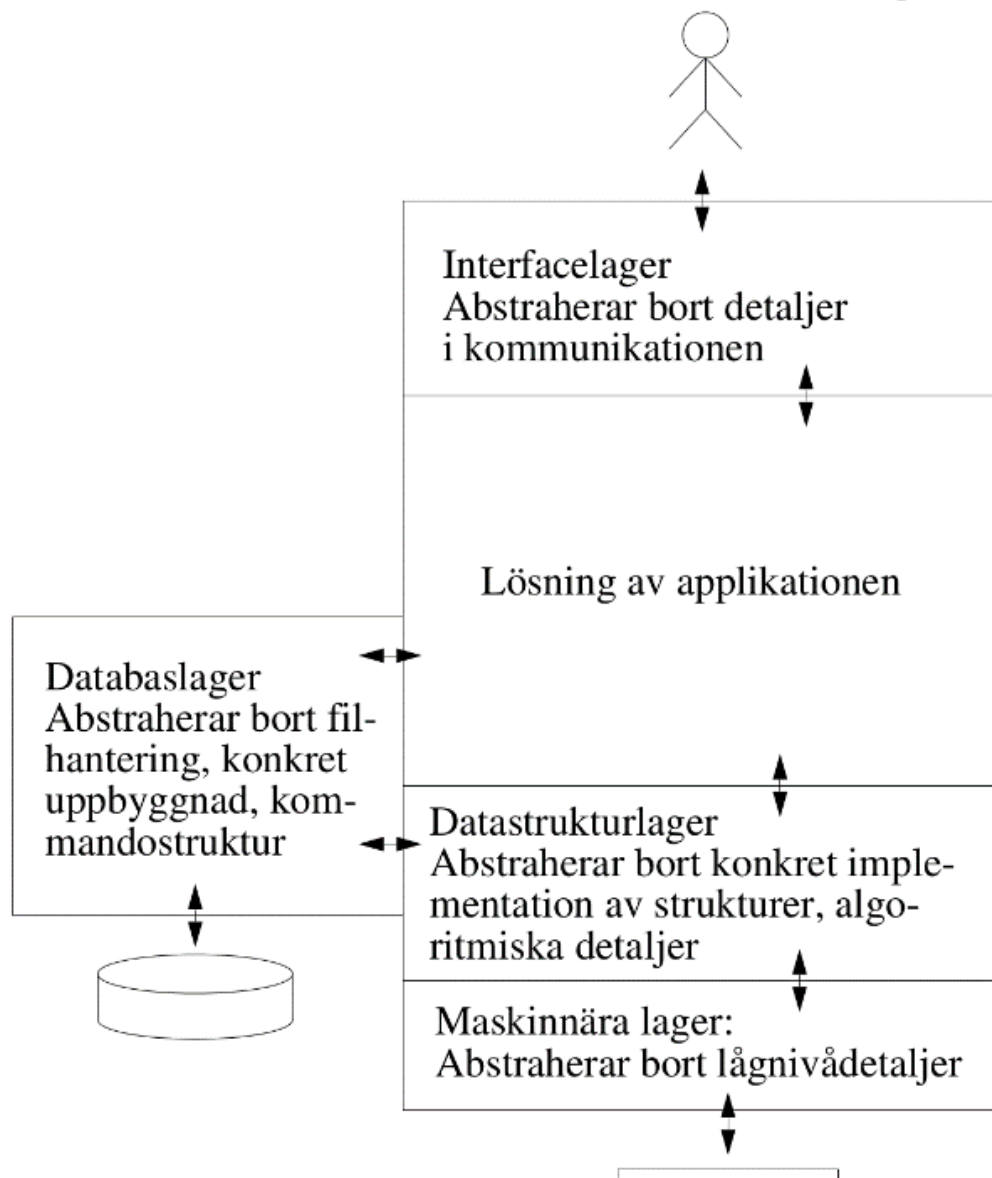
Därför ska den:

- Inte vara alltför stor (eller komplex)
- Ha ett väl avgränsat syfte (dvs. hög "cohesion")
- Ha få relationer till andra moduler (dvs. låg "coupling")

En modul kan vara t.ex:

- En enda subrutin (eller process)
- En databeskrivning (utan algoritmer)
- Ett hopbygge av flera, mindre sådana, om hög "cohesion" råder. *En klass!*

Modul – Abstraktionslager



Modul – Detaljerad design

Förslag på modulattribut som måste specificeras:
(IEEE Standard 1016, från van Vliet)

1. Identifikationen, unikt (dvs. namnet)
2. Typen, alltså t.ex. subsystem, package, klass, fil, subrutin
3. Syfte, övergripande beskrivning
4. Funktionen, relaterat till kravspecifikationen
5. Beståndsdelar, om sådana i sin tur finns
6. Beroenden, relationer till andra komponenter
7. Gränssnitt till andra komponenter, i detalj
8. Externa resurser
9. Utförande, beskrivning av algoritmer, exceptions, etc.. En förfining av funktionen. Motivera väl!
10. Data, beskrivning av representation, format och avsikt med interna data.

Implementation

Begreppet *strukturerad programmering* kan "förklaras" på olika sätt:

- "Programkod ska kunna läsas och förstås i den följd programtexten anger."
- "Använd styrstrukturerna på ett systematiskt sätt" En ingång, en utgång
- "Texten ska återspegla den logiska strukturen"
- "Skriv för folk, inte kompilatorer"

Evaluering av språk

Vilka möjligheter har man att i språket kunna:

- Modularisera på ett vettigt sätt, också "i det stora"?
- Göra egna abstraktioner?
- Införa information hiding?
- Skapa oberoende mellan moduler?
- Skriva läsbar kod?

Hur är det med:

- Ortogonaliteten?
 - Få, kraftfulla konstruktioner som kan kombineras godtyckligt
- Stark typning
 - Gäller "blandning" av typer, typkontroll, när denna äger rum

Utnyttja det språket tillhandahåller!