

Iterativ mjukvaruutveckling

1DV404

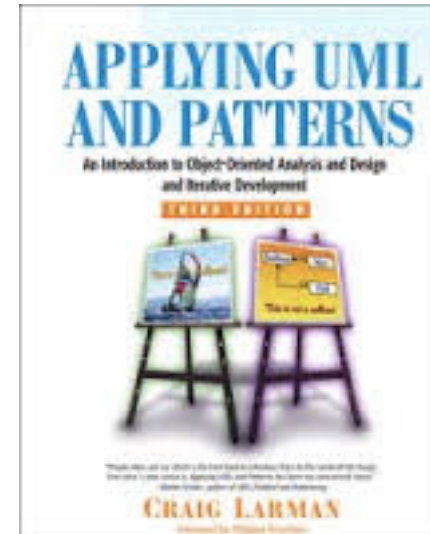
HT14

Jesper Andersson



Om kursen

- ✓ 9-10 föreläsningar
- ✓ Kurslitteratur:
Larman, Craig Applying UML and Patterns,
3rd edition senaste upplagan



- ✓ Kursansvarig och föreläsningar: Jesper Andersson, Jesper.Andersson@lnu.se
- ✓Handledare Emil Carlsson, Oskar Svärd, Johan Widén
- ✓ Examination
Individuella uppgifter och uppgifter i grupp.

Från Kursplanen

Förväntade studieresultat

Kursen syftar till att ge insikt i hur en iterativ utvecklingsprocess för mjukvara ser ut samt ge kunskap i grundläggande arbetsmoment och dokumentation. Efter kursen ska kursdeltagaren ha:

- Grundläggande kunskap i olika mjukvaruutvecklingsprocesser med fokus på iterativa processer.
- Grundläggande kunskap i projektledning av iterativa projekt.
- Grundläggande kunskap om olika roller i ett mjukvaruprojekt.
- Grundläggande förståelse för kravhantering och förmåga att dokumentera krav.
- Grundläggande förståelse för testning av mjukvara och dokumentation kring testning.



Från Kursplanen

Innehåll

Projektledning och projektdokumentation

Versionshantering

Kravhantering och användningsfall

Mjukvaruutvecklingsprocesser med fokus på iterativ mjukvaruutveckling, UP(EDU)

Testning av mjukvara och dokumentation av testning

Spårbarhet.



Mjukvara – från Hantverk till Ingenjörsvetenskap

- ✓ 1940-talet problemlösning genom programmering
- ✓ Komplexiteten ökar
 - Mer komplexa problem
 - Mer komplexa lösningar
- ✓ 1965 Mjukvarukrisen
 - Över budget
 - För sent
 - Dålig kvalitet, mötte inte kraven
 - Svår att hantera
- ✓ Från hantverk till ingenjörskonst! **Eller???**



Mjukvaruutveckling som Hantverk

craft

(krft)*n.*

1. Skill in doing or making something, as in the arts; proficiency.
2. Skill in evasion or deception; guile.
3. **a.** An occupation or trade requiring manual dexterity or skilled artistry.
b. The membership of such an occupation or trade; guild.
4. *pl.* **craft** A boat, ship, or aircraft.

- ✓ Resultatet beror på individens skicklighet
- ✓ Programmeraren – Hantverkaren
- ✓ Lärlingssystem



eller Ingenjörsvetenskap

Engineering

(ɛn'jə-nîr'ing)

n.

1. The application of scientific and mathematical principles to practical ends such as the design, manufacture, and operation of efficient and economical structures, machines, processes, and systems.
2. The profession of or the work performed by an engineer.

- ✓ Förutsägbart
- ✓ Upprepbart
- ✓ Överförbart



Men vad gör vi då? Jo, vi löser problem!

- ✓ Traditionell problemlösningsansats enligt Polya

1. Förstå problemet
2. Arbeta fram en plan för att lösa problemet
3. Genomför planen
4. Utvärdera resultatet

- ✓ Problemlösning med *mjukvara*

- Lösningen tas fram som mjukvara
- Vi använder metoder specifika för mjukvara



Mjukvara – från Hantverk till Ingenjörsvetenskap

- ✓ 1940-talet problemlösning genom programmering
- ✓ Komplexiteten ökar
 - Mer komplexa problem
 - Mer komplexa lösningar
- ✓ 1965 Mjukvarukrisen
 - Över budget
 - För sent
 - Dålig kvalitet, mötte inte kravem
 - Svår att hantera
- ✓ Från hantverk till ingenjörskonst!



Ingenjörsvetenskap

Engineering

(ɛn'jə-nîr'ing)

n.

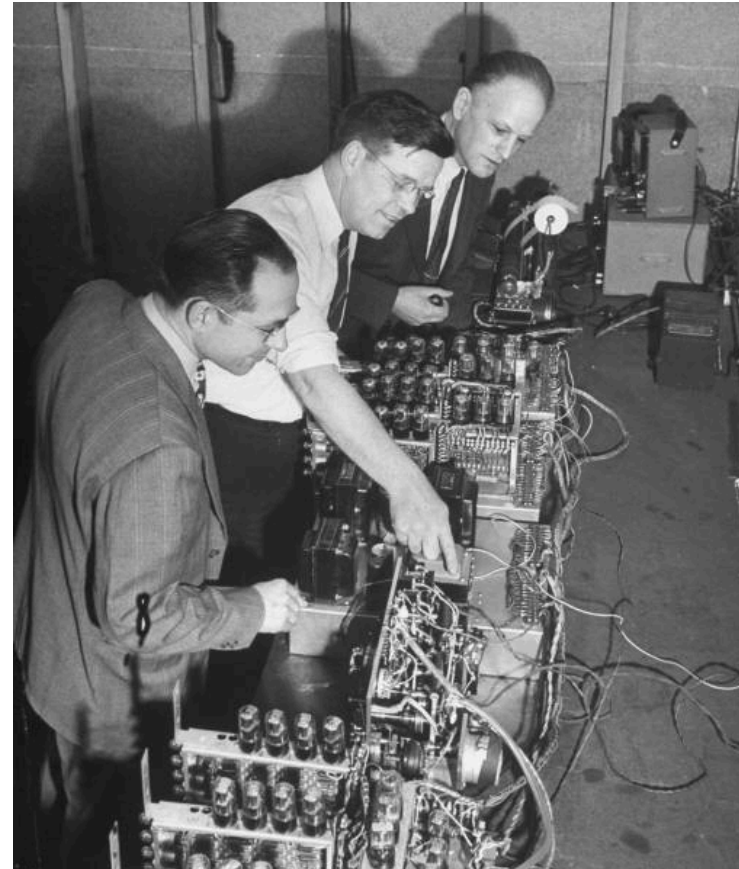
1. The application of scientific and mathematical principles to practical ends such as the design, manufacture, and operation of efficient and economical structures, machines, processes, and systems.
2. The profession of or the work performed by an engineer.

- ✓ Förutsägbart
- ✓ Upprepbart
- ✓ Överförbart



Ingenjörsprinciper

- ✓ Ingenjörer använder sig av vetenskap för att hitta lämpliga lösningar till problem eller förbättra existerande lösningar.
- ✓ Om flera alternativ finns jämför ingenjören alternativen och väljer det som bäst passar kraven.
- ✓ Ingenjörsvetenskap skall vara förutsägbar, dvs ingenjörer försöker förutse hur deras lösning kommer att bete sig i förhållande till dess specifikation innan den har gått i fullskalig produktion



Mjukvaruteknik – Software Engineering

Software engineering (SE) is the profession concerned with specifying, designing, developing and maintaining software applications by applying technologies and practices from computer science, project management, and other fields.

IEEE Software Engineering Body of Knowledge (SWEBOK)

- ✓ Industriell mjukvaruutveckling
 - Metoder
 - Tekniker och
 - Verktyg

- ✓ Programmering i det lilla jmf. med programmering i det stora.



Utvecklingsprojekt – misslyckande

- ✓ Denver Airport – Bagagehantering
- ✓ London – Ambulansdispatcher

9/9

0800 Andam started

1000 . stopped - andam ✓ { 1.2700 9.037 847 025
1300 (032) MP-MC 2.130476415 (2) 4.615925059(-2) 9.037 846 995 correct


(033) PRO 2 2.130476415
correct 2.130676415

Relays 6-2 in 033 failed special speed test
in relay "no test"

Relays changed

1100 Started Cosine Tape (Sine check)

1525 Started Multi-Adder Test.

1545  Relay #70 Panel F
(moth) in relay.

First actual case of bug being found.

1630/1630 Andam started.

1700 closed down.

Relay 3145
Relay 3370



Orsak?



Kommunikation

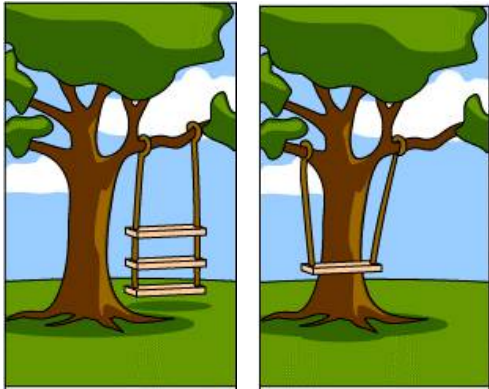
Ett typiskt utvecklingsprojekt

- Hur kunden förklarade det?



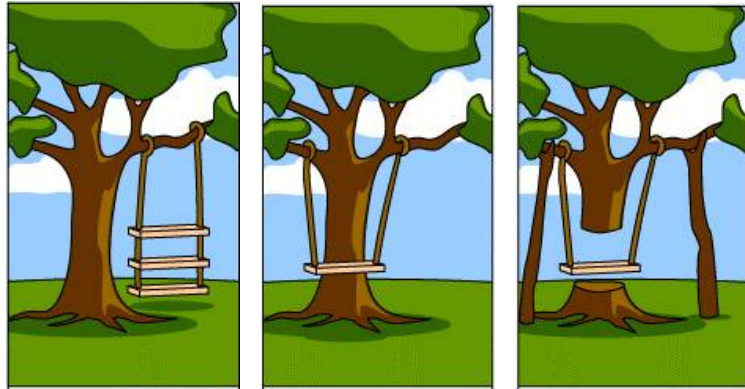
Ett typiskt utvecklingsprojekt

- Hur projektledaren förstod det?



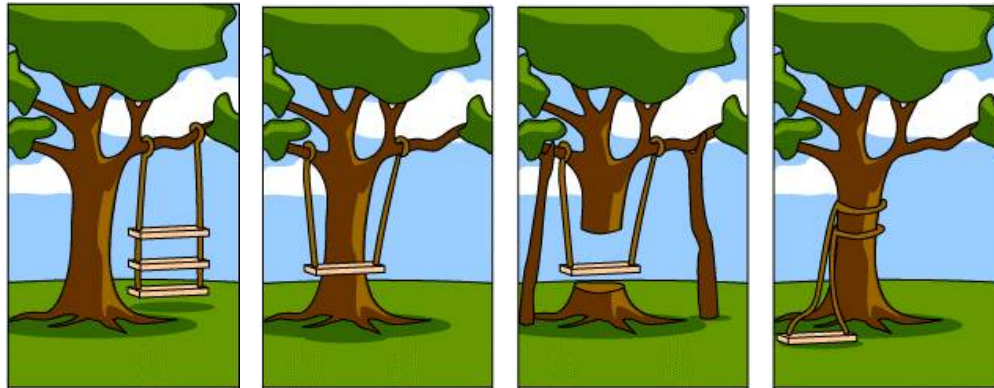
Ett typiskt utvecklingsprojekt

- Hur arkitekten designade systemet?



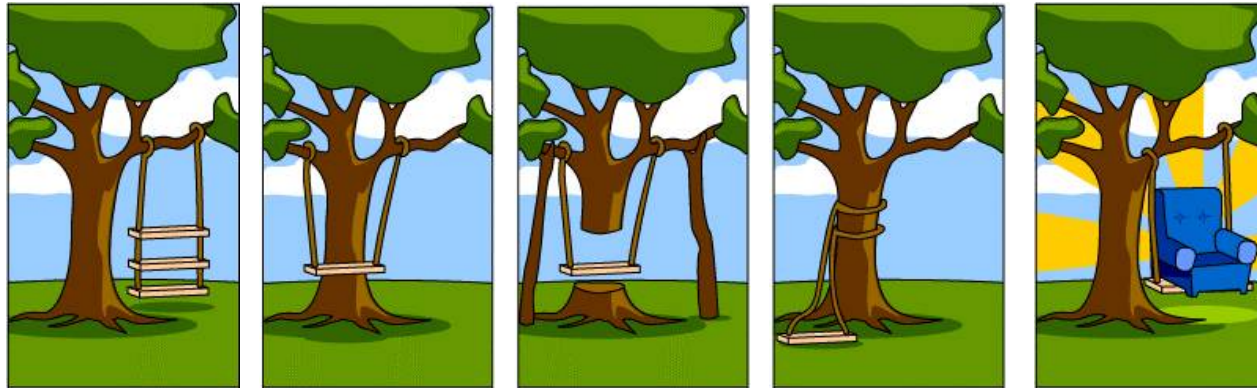
Ett typiskt utvecklingsprojekt

- Hur utvecklarna implementerade det!



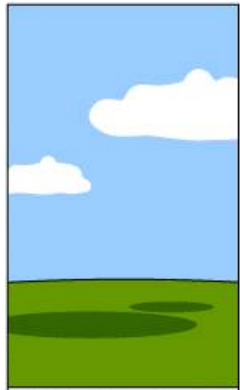
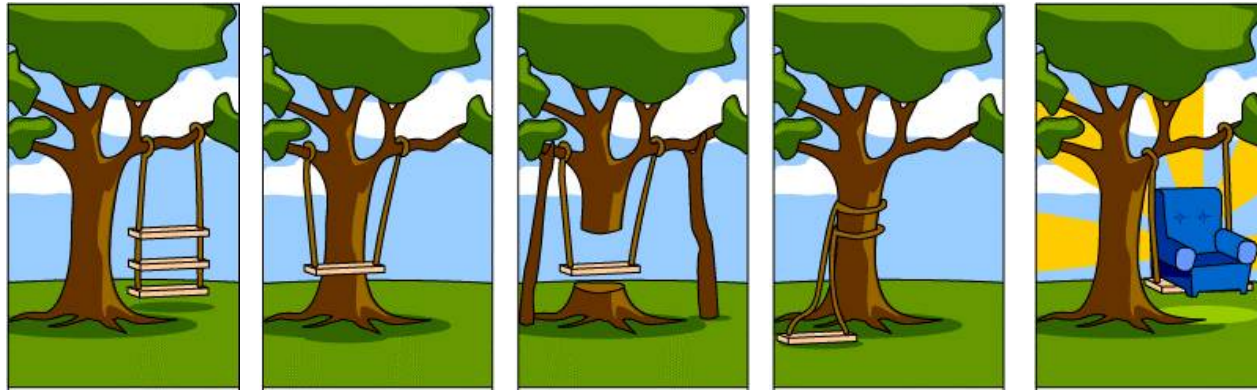
Ett typiskt utvecklingsprojekt

- Hur det marknadsfördes!



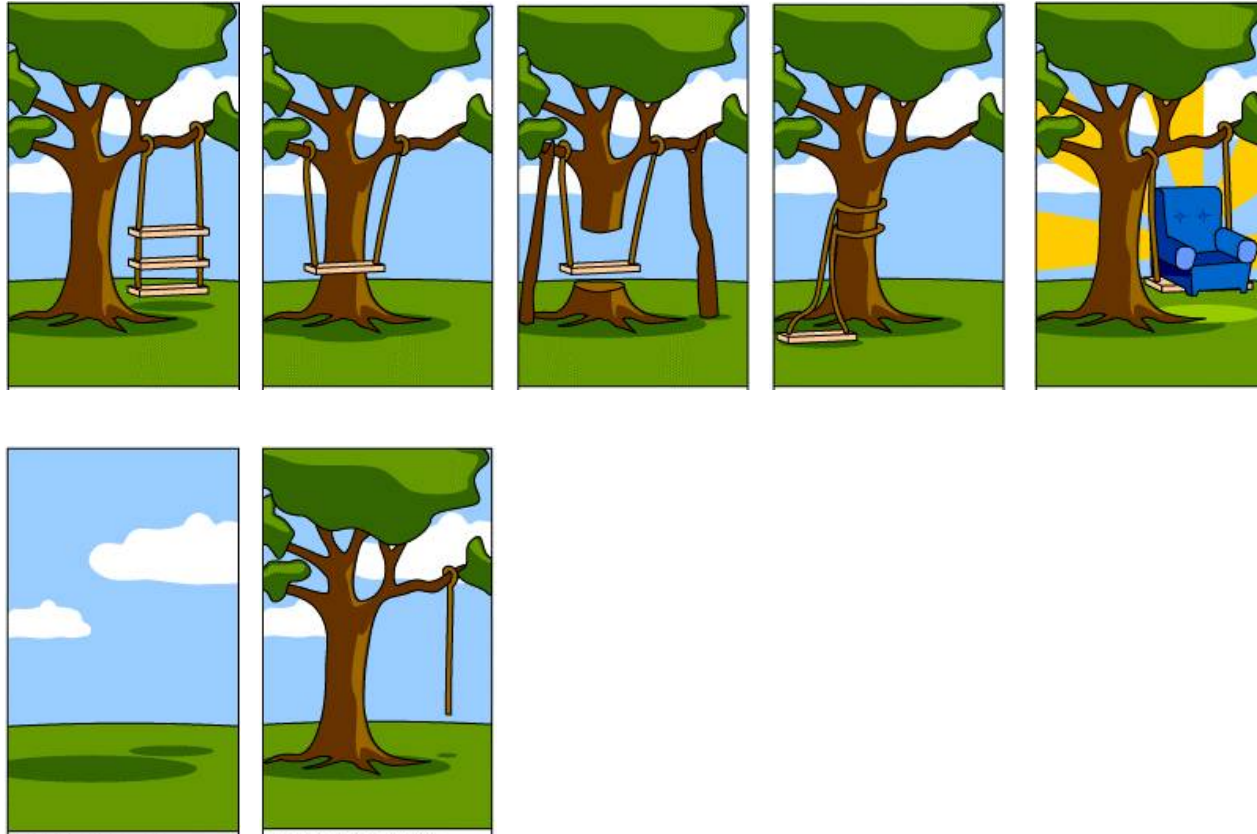
Ett typiskt utvecklingsprojekt

- Hur projektet dokumenterades?



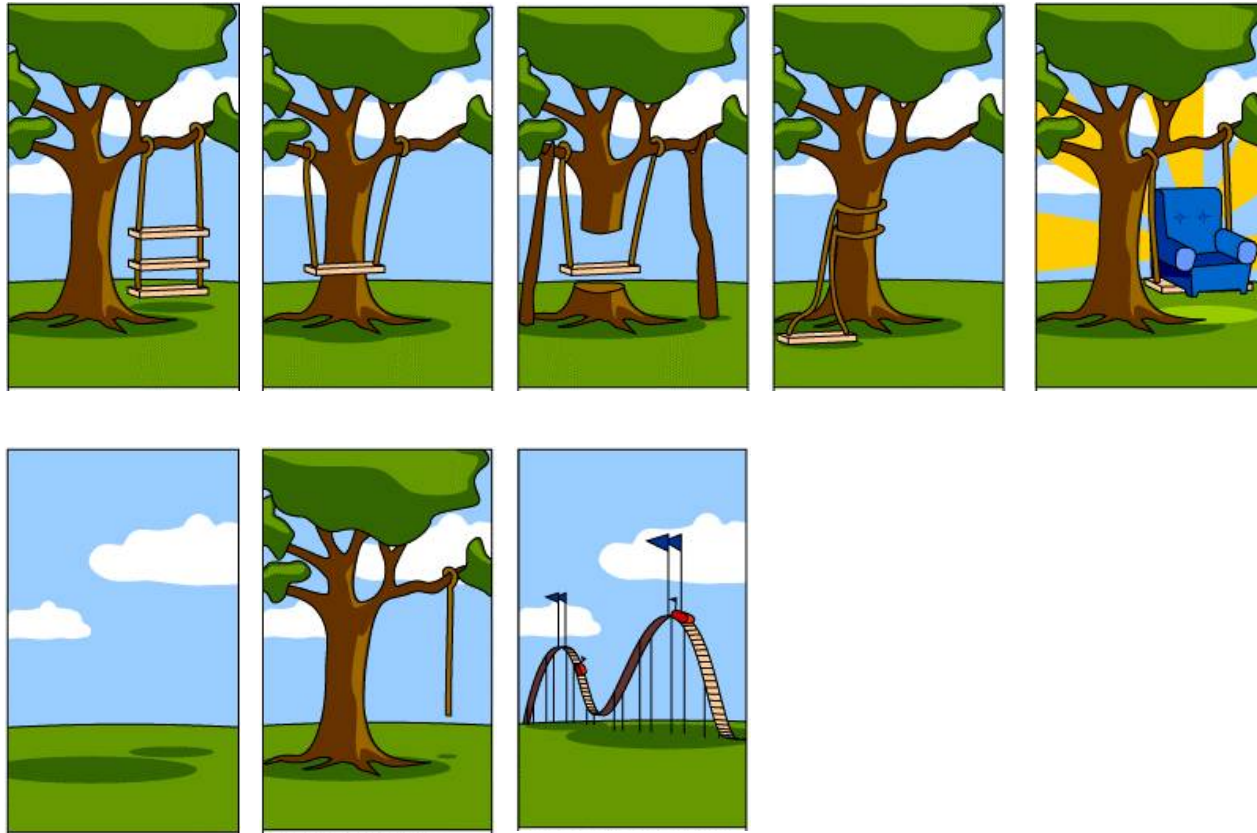
Ett typiskt utvecklingsprojekt

- Vad som levererades till kunden?



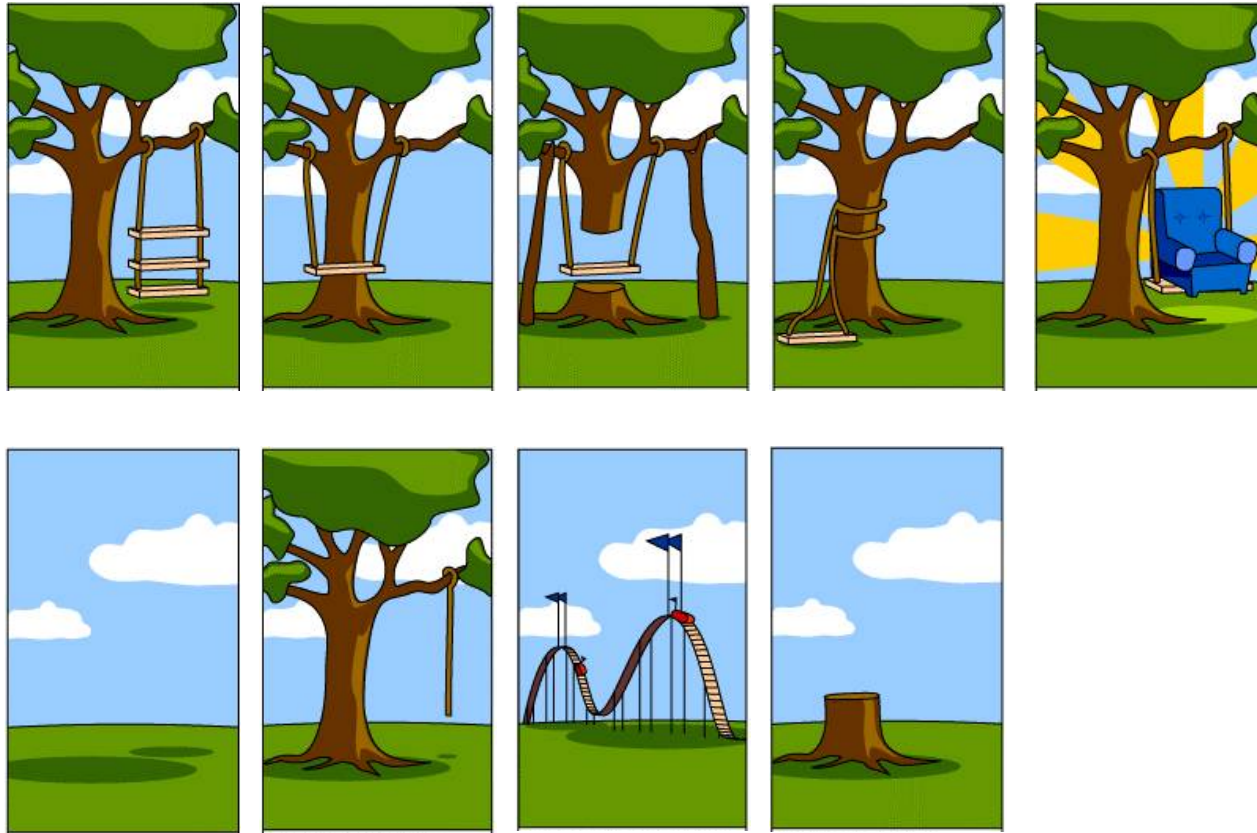
Ett typiskt utvecklingsprojekt

- Vad kunden fakturerades för!



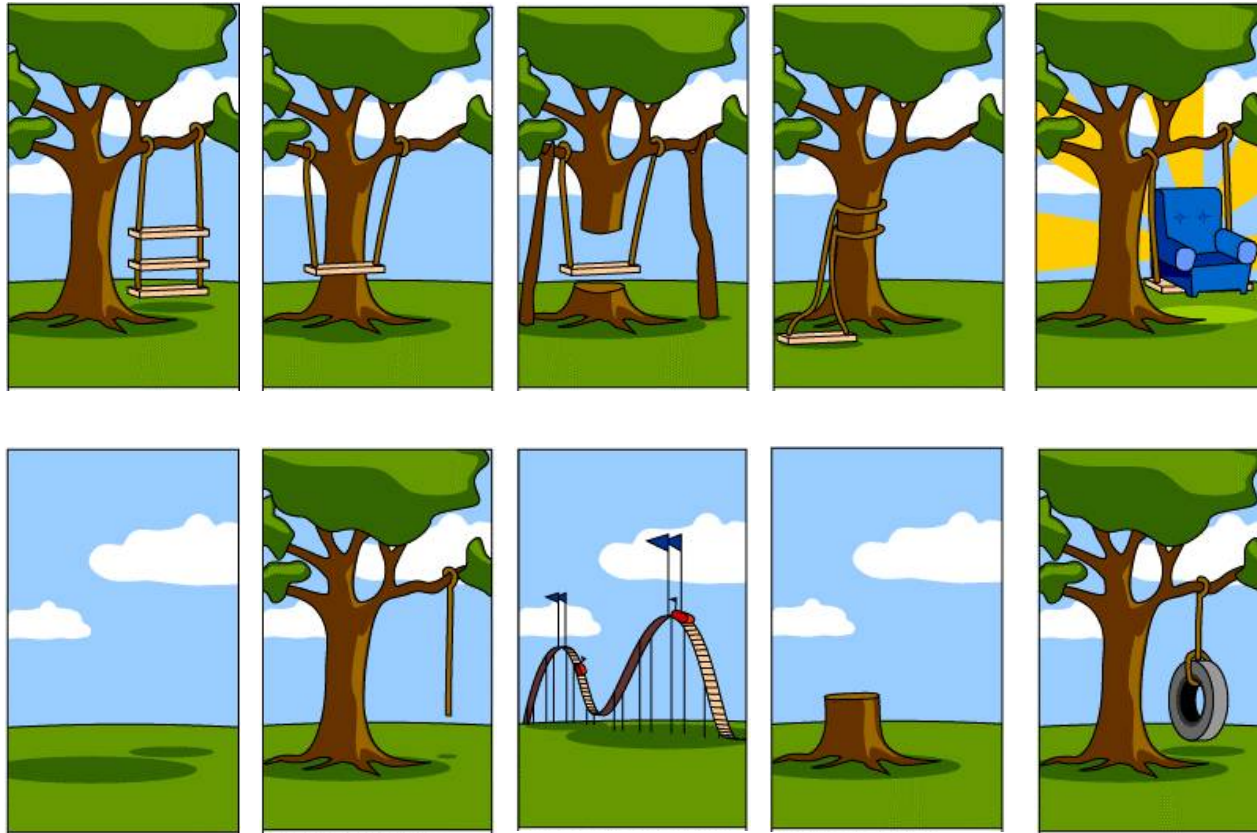
Ett typiskt utvecklingsprojekt

- Hur systemet supportades efter leverans?



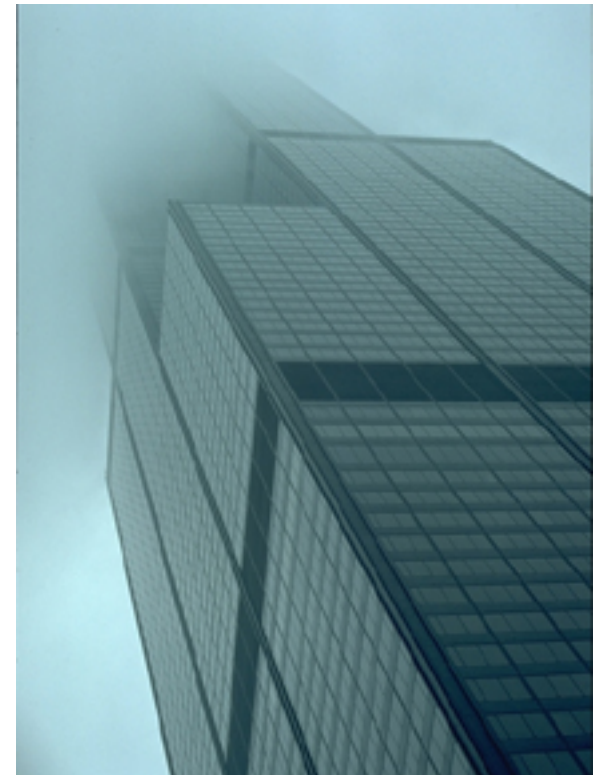
Ett typiskt utvecklingsprojekt

- Vad kunden egentligen behövde!



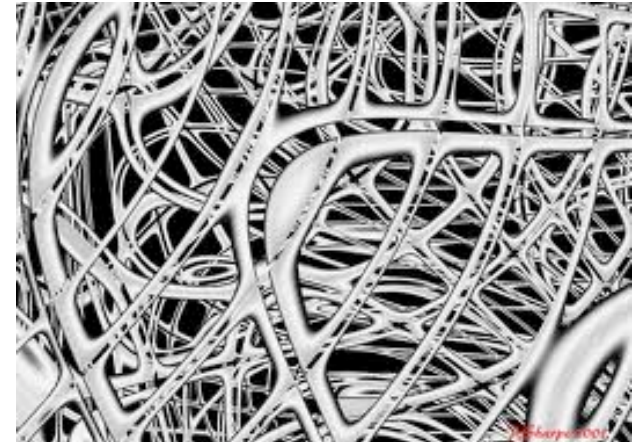
Mjukvara – Egenskaper som gör utveckling till en rejäl utmaning.

- ✓ Den inbyggda *komplexitetet* som växer mångfalt snabbare än antalet rader kod.
- ✓ *Förändringbarheten* som kommer av att mjukvaran är en modell av verkligheten och verkligheten förändras.
- ✓ *Anpassningsbarhet* som kommer sig av att mjukvaran samverkar med omgivande system
- ✓ *Osynlighet* som kommer av att vi inte fullständigt kan visualisera alla aspekter av mjukvaran.



Hur hanterar vi komplexitet?

- ✓ Abstraktion
 - Förenklingar
 - Beskrivning av element m.h.a nödvändiga egenskaper (beroende på sammanhang).
 - Exempel, Dataabstraktion, Instruktionsabstraktion
- ✓ Nedbrytning
 - Problem bryts ned i delproblem, som i sin tur bryts ned i ytterligare delproblem...
 - Problemet delar upp lösningen i delar
 - Divide n' conquer

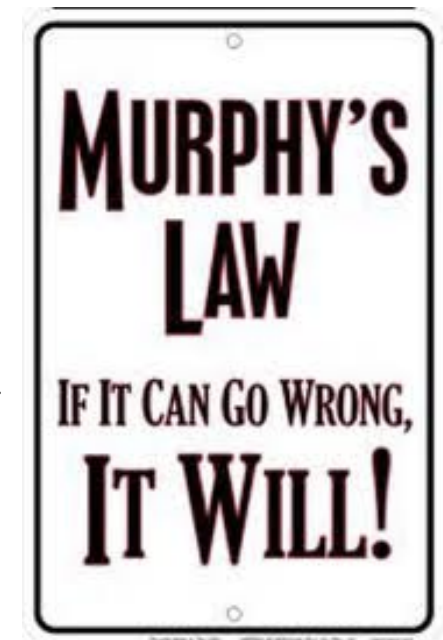


Hur hanterar vi förändringsbarhet?

“If you are not moving at the speed of the marketplace you’re already dead – you just haven’t stopped breathing yet”

Jack Welch, CEO, GE , 1981-2001

- ✓ Mjukvaruprocesser
 - Struktur
 - Roller och Ansvar
- ✓ Mjukvarumodeller
 - Begriplighet
 - Förståelse
- ✓ Riskhantering ... om något händer så har vi planerat för det.



Hur hanterar vi osynligheten?

- ✓ Modeller

- För krav
- För design
- För implementationer
- För installation
- För operation

- ✓ Modeller underlättar kommunikation!



- ✓ Vi får ett *gemensamt* språk med *vokabulär*, *syntax* och *semantik*

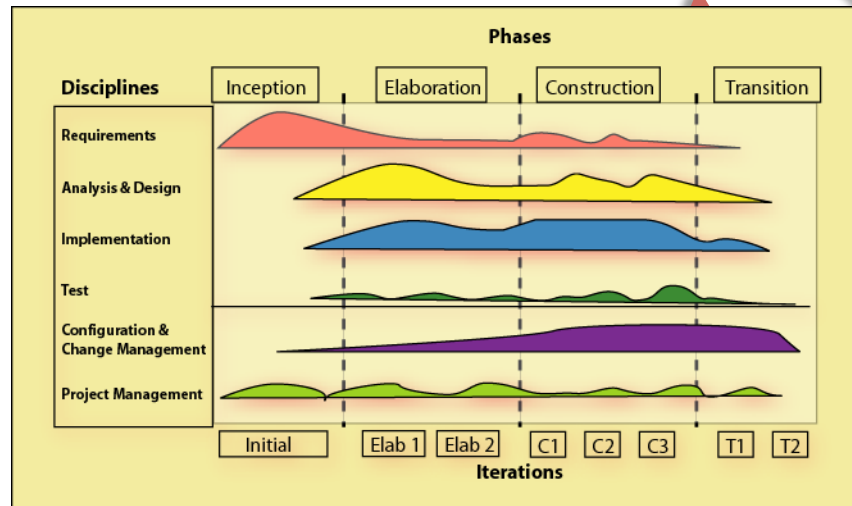
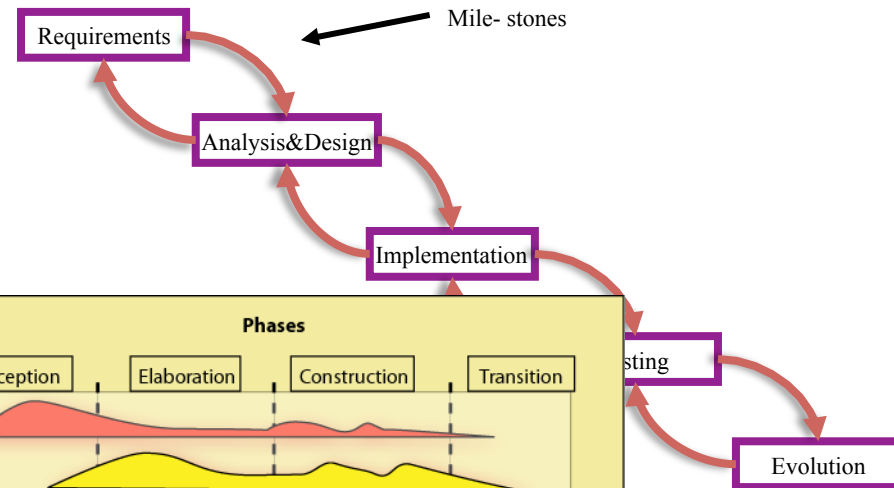
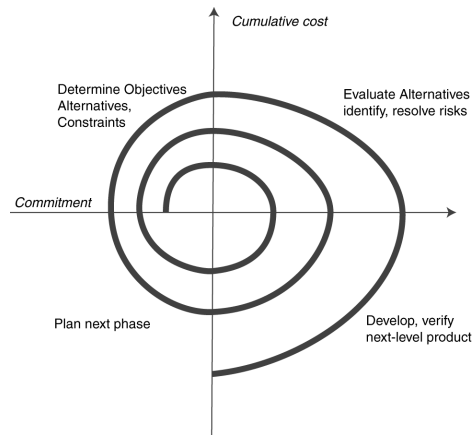
Processaktiviteter

- ✓ Krav – Requirements Engineering
- ✓ Design – Software Design
- ✓ Test – Software Testing
- ✓ Konfigurationshantering – Configuration Management
- ✓ Implementation
- ✓ Projektledning – Project management
- ✓ Kvalitetshantering – Software Quality Management
- ✓ Produkthantering – Software Product Management



Strukturera aktiviteterna – en Process

- ✓ Waterfall – Royce (1970-talet)
- ✓ Spiral – Boehm (1980-talet)
- ✓ Unified Process (1990-talet)
- ✓ Agile Processes (2000-talet)



Vattenfallsmodellerna (1970)

MANAGING THE DEVELOPMENT OF LARGE SOFTWARE SYSTEMS

Dr. Winston W. Royce

INTRODUCTION

I am going to describe my personal views about managing large software developments. I have had various assignments during the past nine years, mostly concerned with the development of software packages for spacecraft mission planning, commanding and post-flight analysis. In these assignments I have experienced different degrees of success with respect to arriving at an operational state, on-time, and within costs. I have become prejudiced by my experiences and I am going to relate some of these prejudices in this presentation.

COMPUTER PROGRAM DEVELOPMENT FUNCTIONS

There are two essential steps common to all computer program developments, regardless of size or complexity. There is first an analysis step, followed second by a coding step as depicted in Figure 1. This sort of very simple implementation concept is in fact all that is required if the effort is sufficiently small and if the final product is to be operated by those who built it – as is typically done with computer programs for internal use. It is also the kind of development effort for which most customers are happy to pay, since both steps involve genuinely creative work which directly contributes to the usefulness of the final product. An



Processmodeller – “Drivkrafter”

- ✓ Milstolpar
- ✓ Iterationer
- ✓ Inkrement
- ✓ Risker



Milstolpar

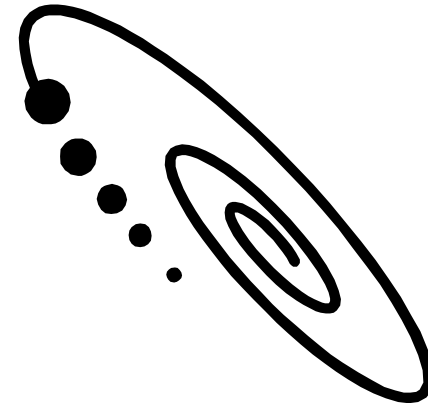
- ✓ Dokumentdriven, dokumentet är klart ...
- ✓ Releasedriven, “releasen” är klar ...

- ✓ **Perspektiv:** Vem gör vad och när är det klart!!!!

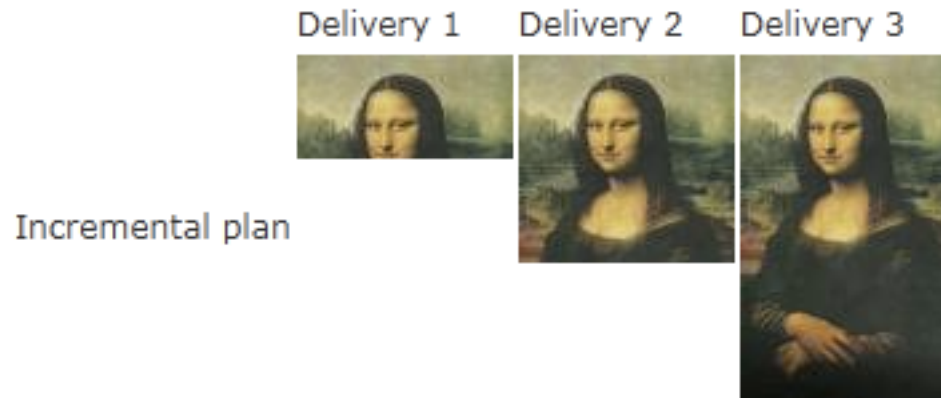


Iterationer & Inkrement

- ✓ Enklare att planera (!?)
 - Mindre steg
 - Kortare tid
 - Enklare att verifiera och validera
 - Snabbare återhämtning om vi gjort fel eller gått fel väg.



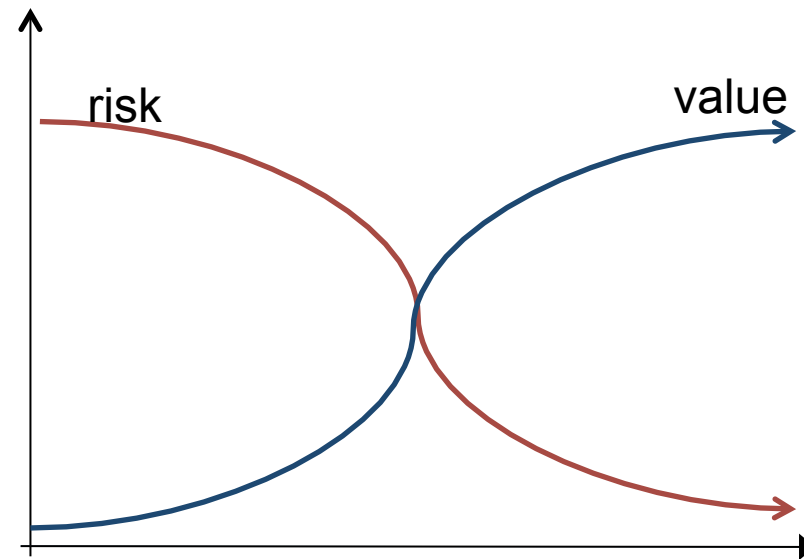
Iterativ vs. inkrementell ansats



Källa: <http://www.applitude.se/>

Riskhantering

- ✓ **Proaktiva** aktiviteter för att **minimera osäkerheter** och **eventuell skada** kopplat till ett utvecklingsprojekt.
- ✓ Beslut baserad på fakta och kunskap inte gissningslekar.
- ✓ Genomförs kontinuerligt under systemets livslängd
- ✓ Några riskkategorier
 - Produktens storlek,
 - Marknad,
 - Kunder,
 - Process,
 - Utvecklingsteknologier,
 - Personal (antal & erfarenhet),
 - Planer
 - Kostnader



Utmaningar i ett utvecklingsprojekt

- ✓ Ge exempel på utmaningar med
 - Planering
 - Genomförandet
- av ett mjukvaruprojekt



Idag

- ✓ Lite kring mjukvaruutvecklingens historia
- ✓ Problematiken kring ett projekt
- ✓ Aktiviteter när vi utvecklar mjukvara
- ✓ Processen – hur vi strukturerar aktiviteter
- ✓ Drivkrafter – varför vi väljer en viss struktur.



På torsdag

- ✓ Mer om utvecklingsprocesser
 - ✓ Aktiviteter
 - ✓ Roller
- och sist men inte minst
- ✓ Planering

