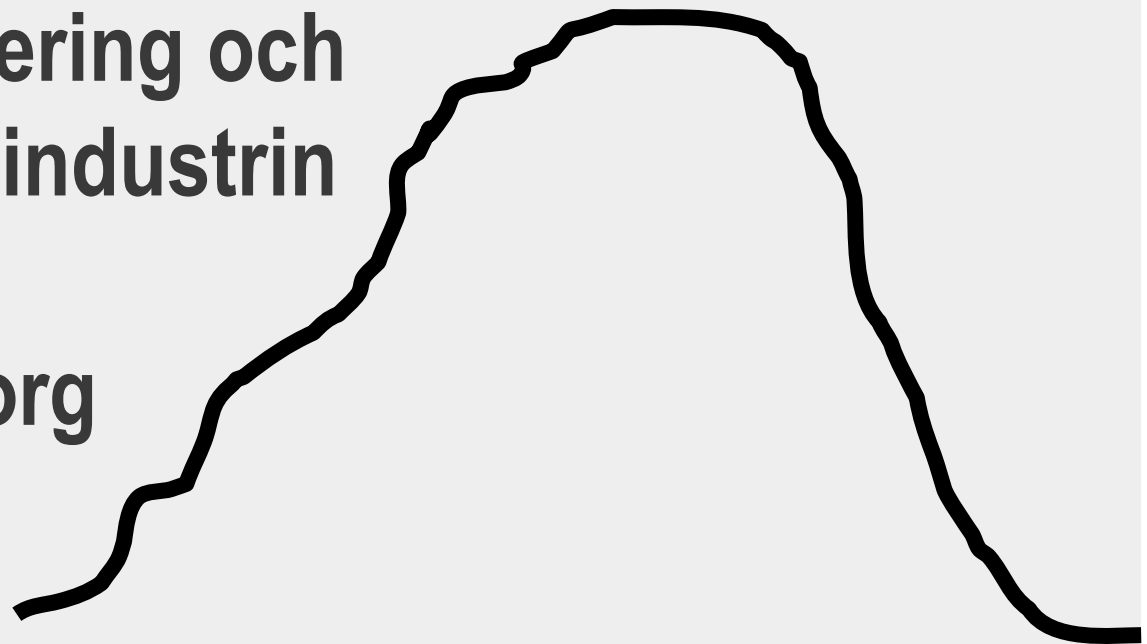




Krav och trender

Praktisk kravhantering och
annat nyttigt från industrin

Christian Ehrenborg



Presentation



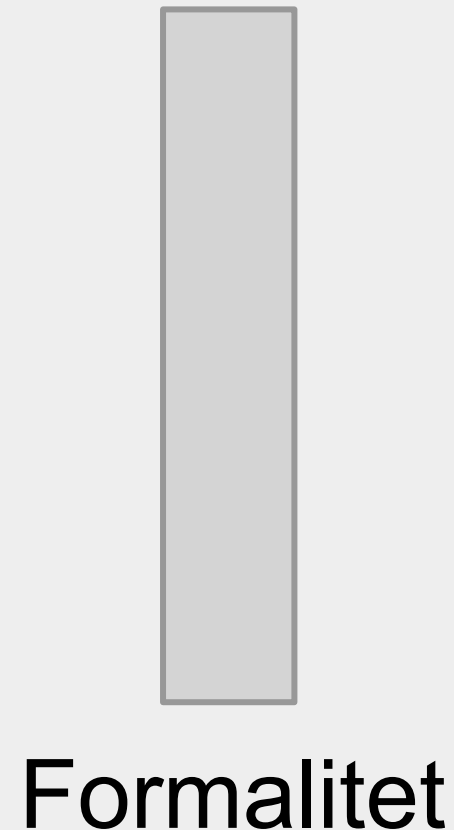
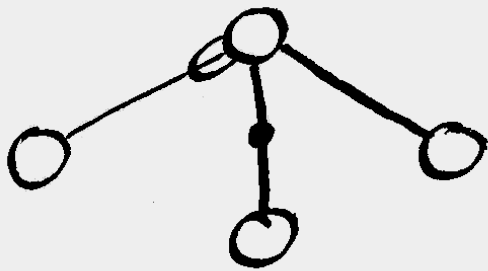
Christian Ehrenborg

- D79
- Konsult sen 1988
- Utvecklade RUP
- Har arbetat med: Telefoner, Tåg, Administration, Web, Ledningssystem, flygplatser...
- Har jobbat med de flesta kravmetoder

Hur det hela började

Krav hantering på 80 talet

- Mil. standards
- ERM – data modellering
- Krav i tjocka buntar



Forntiden

90 talet

- **Objectory - RUP**
- **Rapid Prototyping som underström**
- **Komplexiteten ökade**
- **Metodgurus tog över**

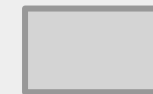
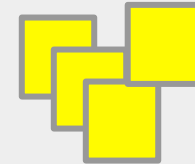


Formalitet

Nyss

00 talet

- Scrum
- User Stories
- Man sökte enkla lösningar
- Pendeln slog åt andra hållet

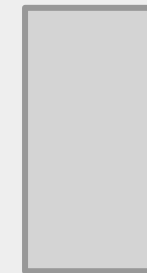


Formalitet

Nu

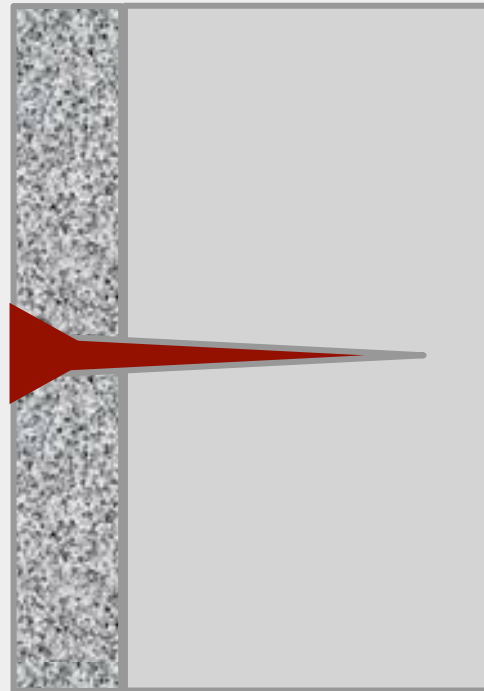
10 talet

- **Brytningstid**
- **Allt är User Stories**
- **Metodtrötthet**



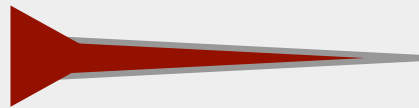
Formalitet

Men vad är problemet?



En Skruv som skall skruvas ut

Metod-religiösa vet alltid svaret



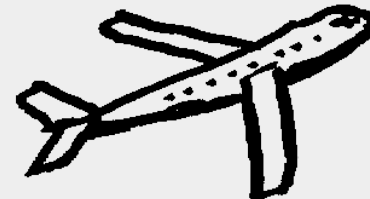
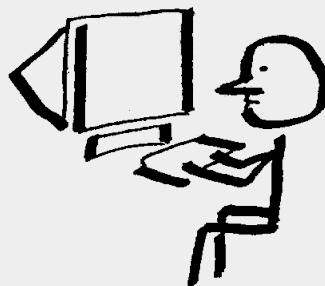
Som att diskutera vilken
skruvmejsel som skall användas

Kolla först



Vad skall byggas?

- **Web information**
- **Fysiska produkter/Spel**
- **Förvaltande (verksamhetsstöd)**
- **Kritiska, regulativa krav**



Ökande formalism

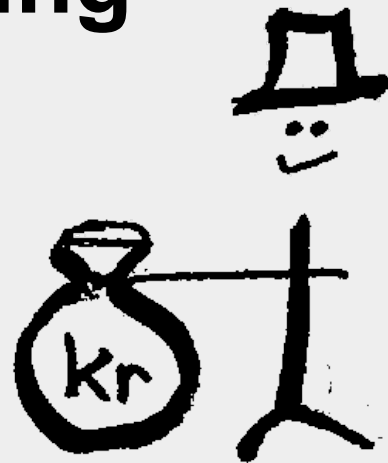
Underhållstid?

- **1 månad – sen är det över**
- **6 månader**
- **2 år**
- **30 år – tåg, telefonstationer, administrativa system...**

Ökande underhållskrav

Kontraktrelation?

- Egen utveckling
- Inhyrda utvecklare
- Kontraktsutveckling
- Fast pris



Ökande formalism

Kundens förtroende för utvecklarna

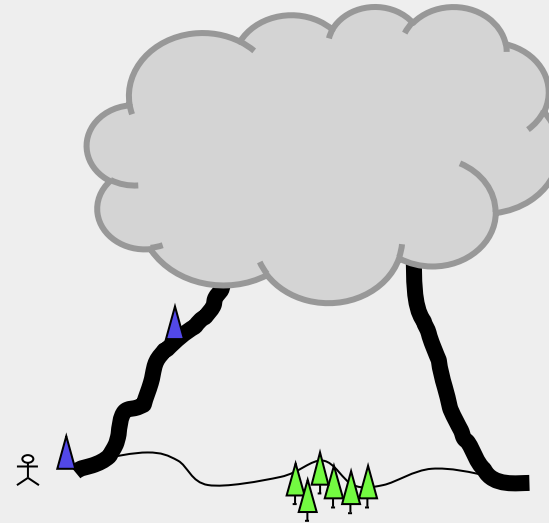
- **Känner kunden innan och utan**
- **Viss branch erfarenhet**
- **Har aldrig sysslat med detta segment**
- **Tillhör en annan kultur**

Ökande formalism

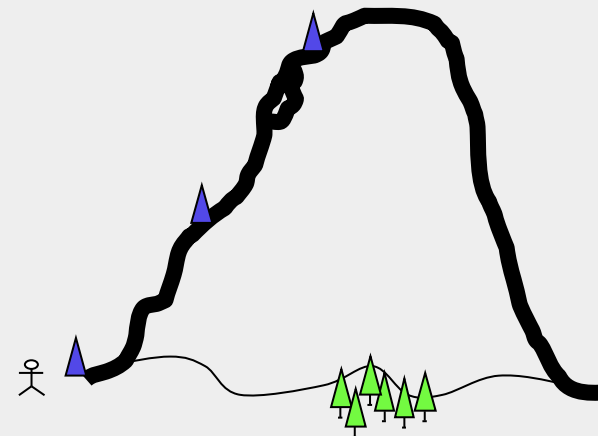


Kan man se målet för projektet?

Målet är dolt i
dimma.



Vi vet vad målet är, och
det går att se



Verktyg för krav

Glöm grafiska metoder. Krav är text.

- **Kvalitets attribut**

- **Enkelt i teorin**

- **Svårt i många projekt**

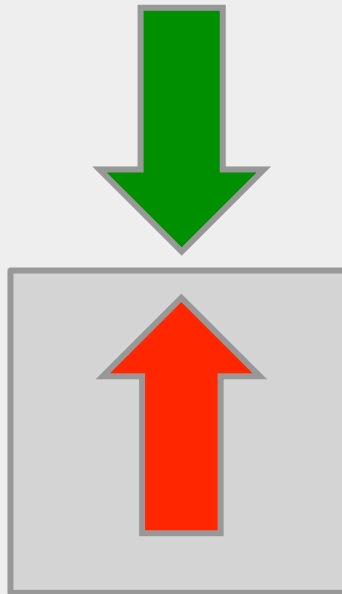


- **Funktionsorienterade krav**

Funktions orienterade krav

Två olika aspekter

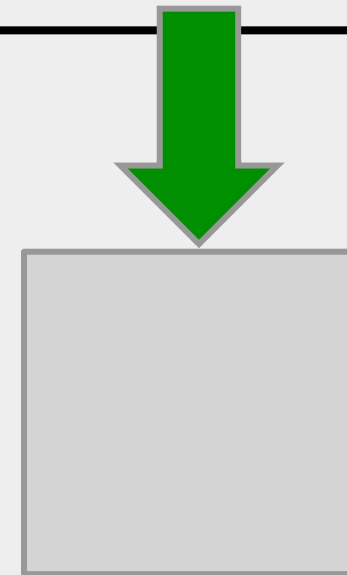
Förlopps beskrivningar



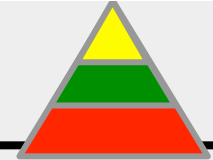
Funktions beskrivningar

Exempel

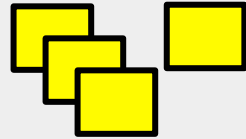
Report incident
Add more facts to an
existing report
Join reports into a case



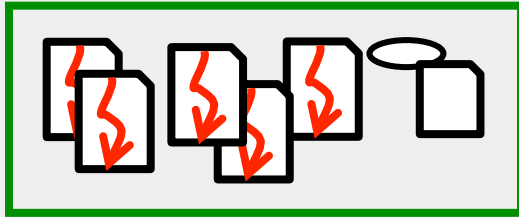
Create report
Update report
Delete report



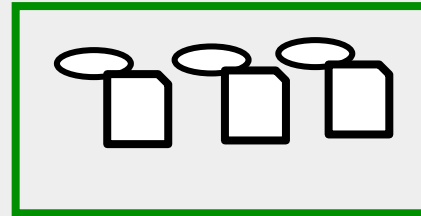
De tre kravnivåerna



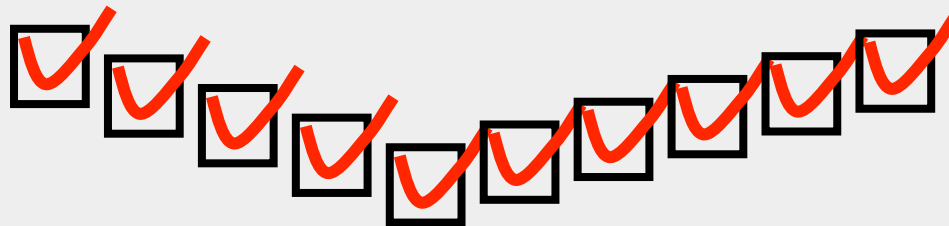
Behov



or

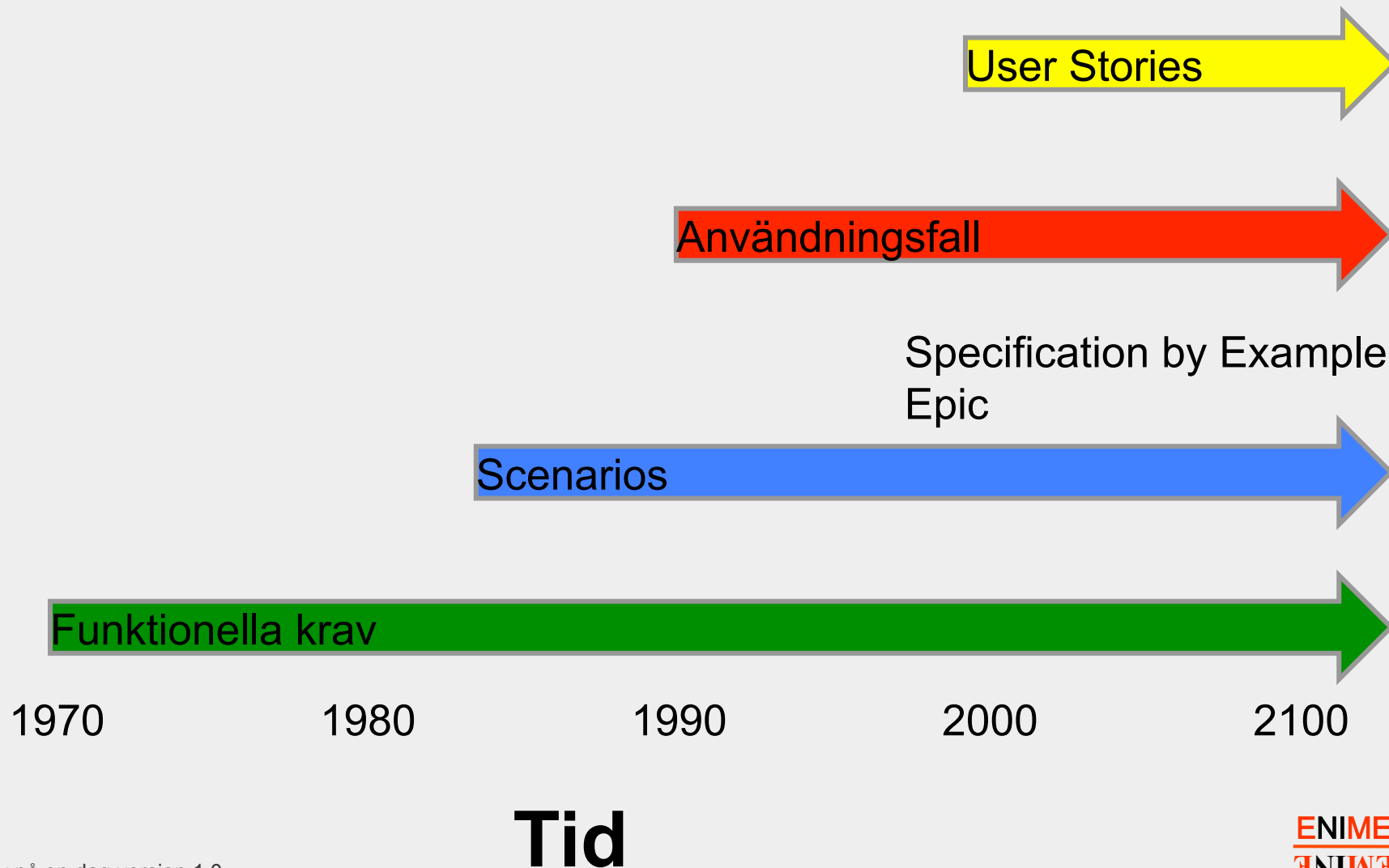


Koncept

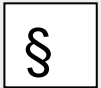


Funktioner

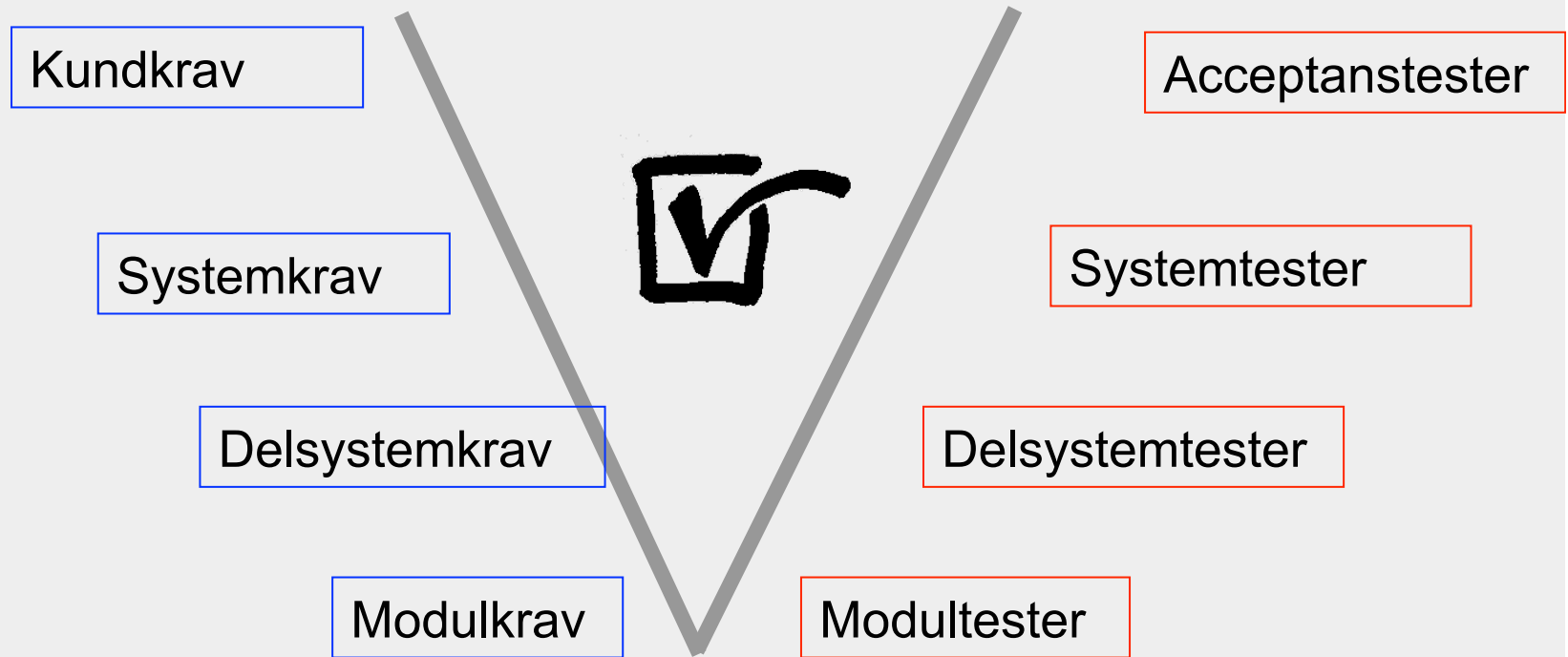
Utvecklingen av kravmetoder



Funktionella krav

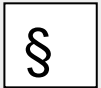


Kravnedbrytning – och V-modellen
I nedbrytningen ligger ett arkitektur ansvar

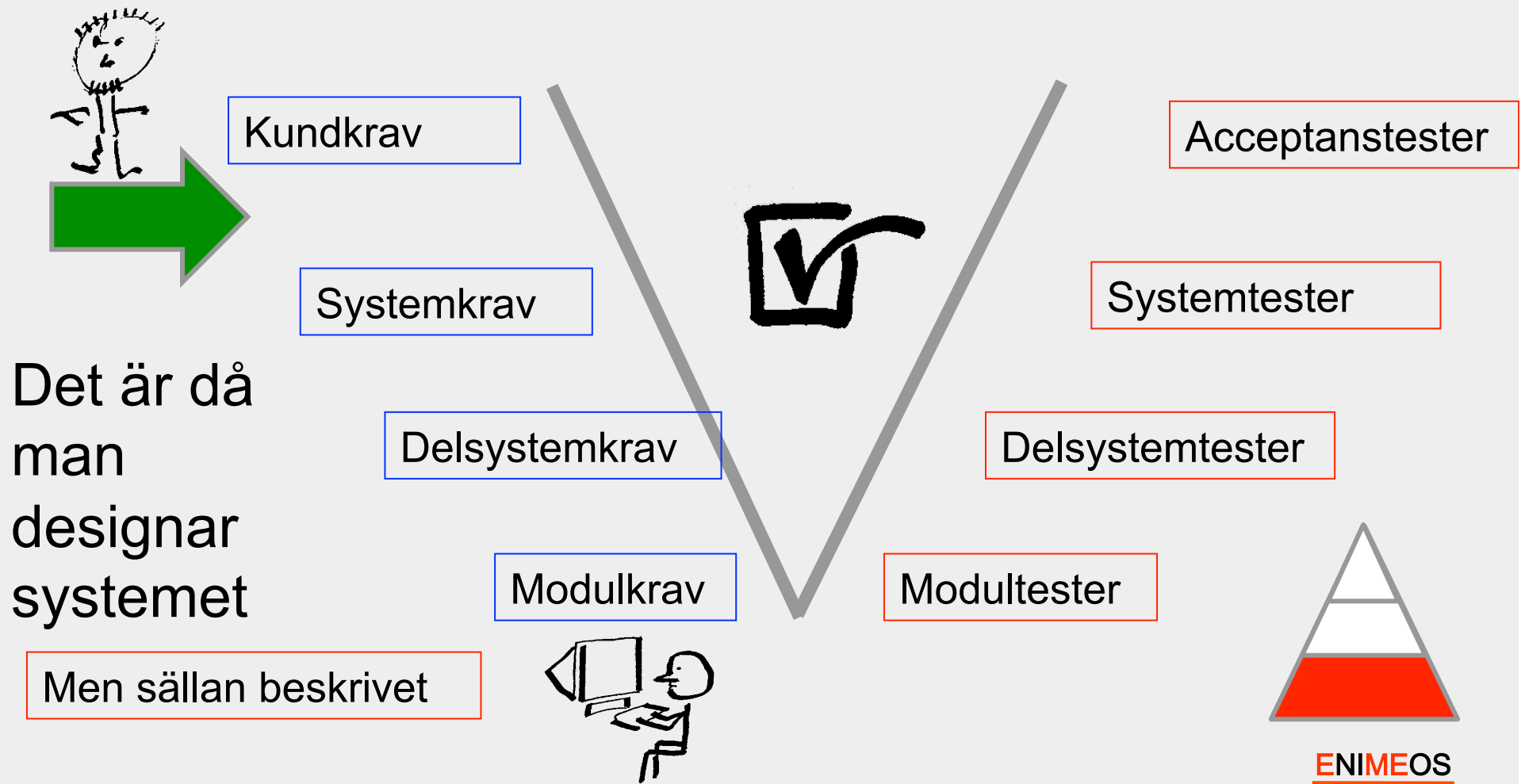


**Den som realiserar en modul har
inget ansvar att helheten fungerar!**

Var ligger det kreativa arbetet?



Det kreativa arbetet ligger i nedbrytningen



Scenarios



Report incident
Report traffic incident
Report burglary



C:a 1 sida text per scenario

Nästa steg är att bryta ner scenarierna i funktionella krav

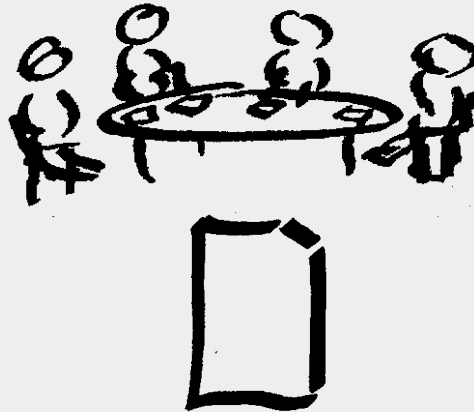


Den som realiserar ett scenario har ansvar att systemet fungerar!

Var ligger det kreativa arbetet?

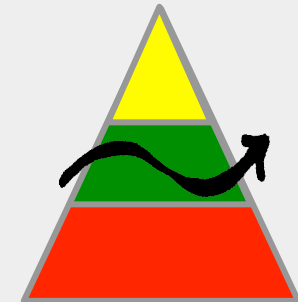


Användare

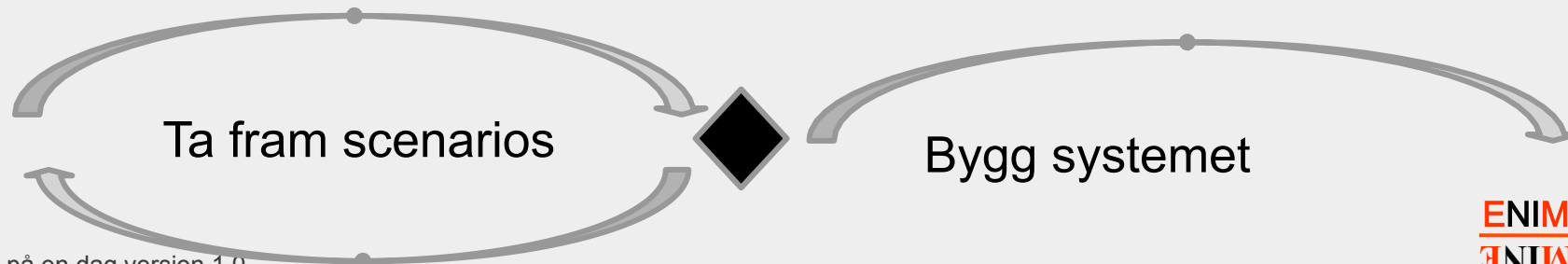


Utvecklare

Testare



Att gemensamt beskriva förloppen
– det är då systemet utformas



Användningsfall

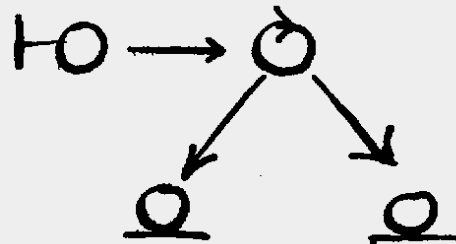


- Report incident
- Add more facts to an existing report
- Join reports into a case



C:a 5 sidor text per användningsfall

Nästa steg är att bryta ner användningsfallen i funktionella krav



Var ligger det kreativa arbetet?

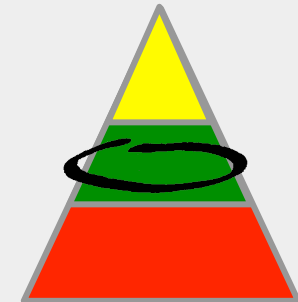
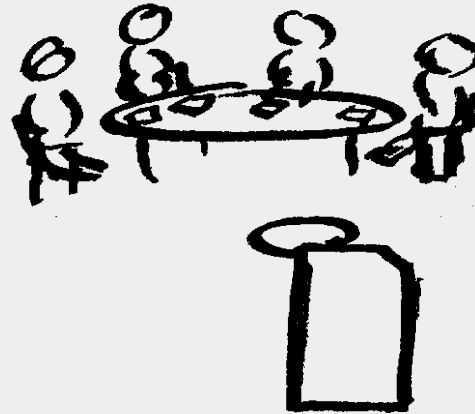


Användare?

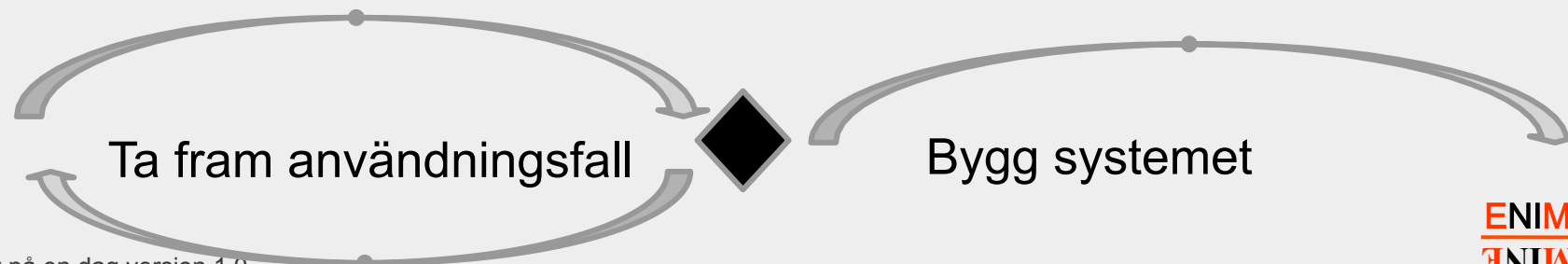
Kravare

Utvecklare?

Testare?






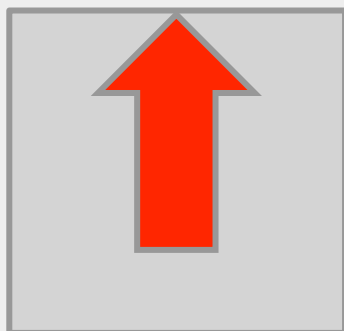
Att gemensamt beskriva förloppen
– det är då systemet utformas






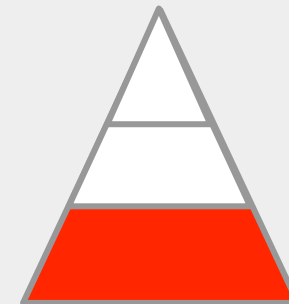
Missbruk av användningsfall (och scenarios)



-  Report incident
-  Add more facts to an existing report
-  Join reports into a case



-  Create report
-  Update report
-  Delete report



Användningsfall eller scenarios



Användningsfall beskriver de förlopp som skall finnas i systemet. **Inget mer inget mindre.**



Scenarios är exempel på de förlopp som skall finnas i systemet. **Många fler kan finnas.**

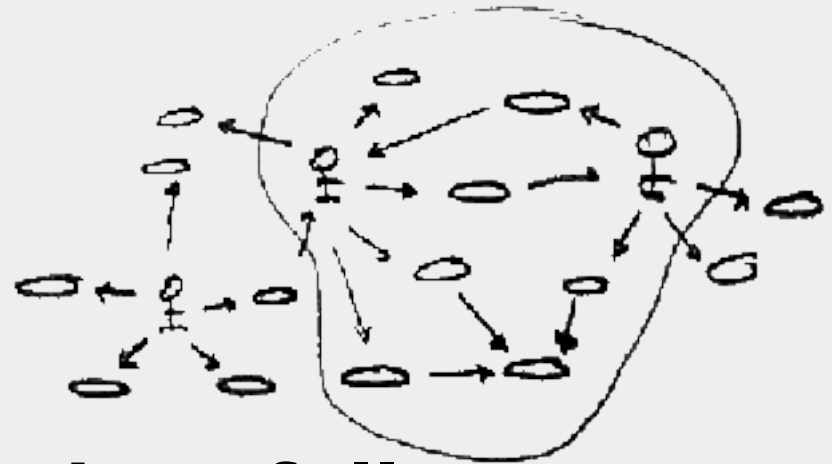
Utvecklingen av användningsfall

1.0 Aktörer, beskrivning, uses & Extends

+ Extension point

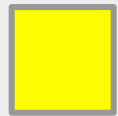
+ Pre & post villkor

+ Arv mellan användningsfall



2.0 Aktörer, beskrivning

User Stories



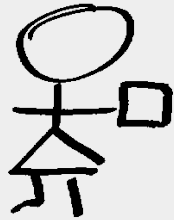
As a officer I need to file reports



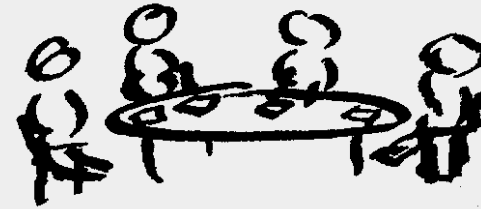
C:a 1 mening per User Story

Nästa steg är att bygga nått som användarna gillar...

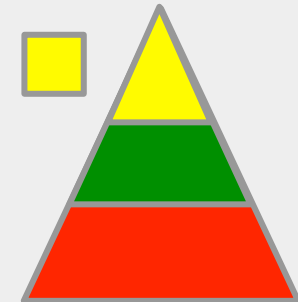
Var ligger det kreativa arbetet?



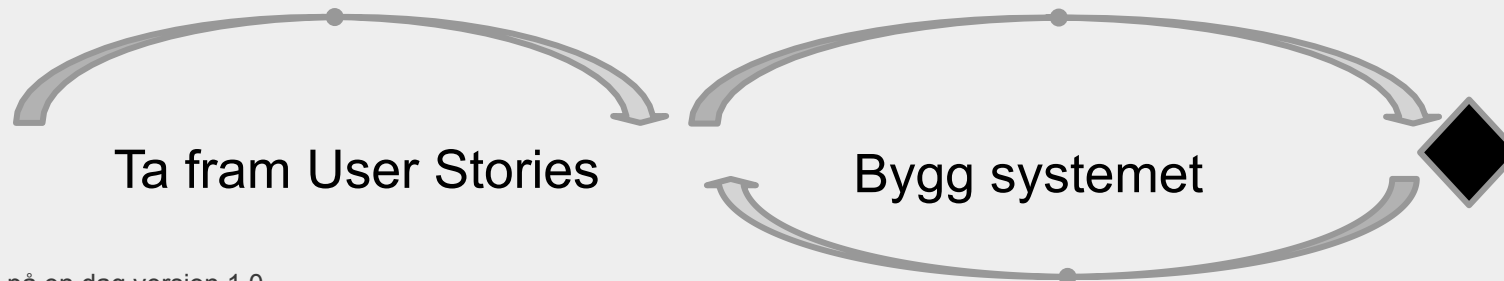
Produkt ägare



Utvecklare



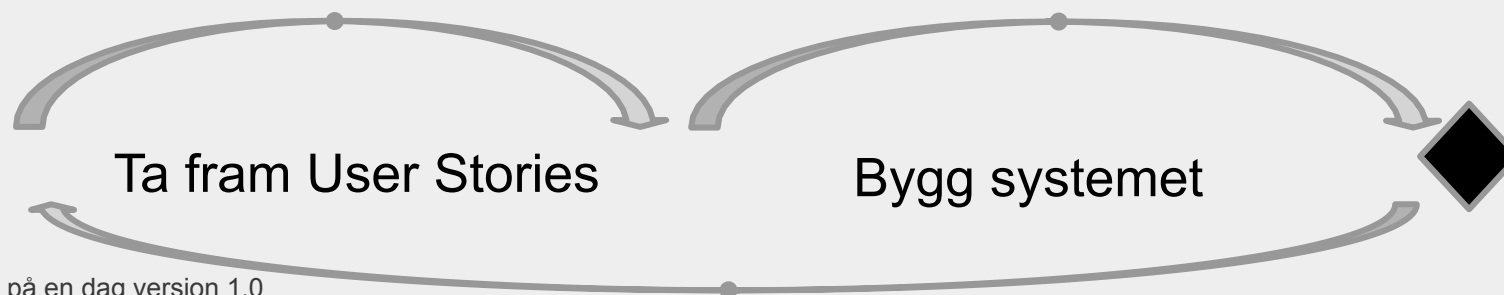
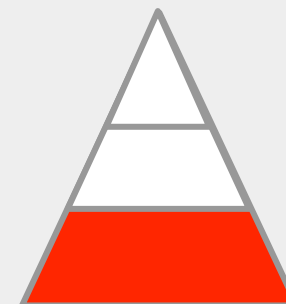
Att hitta en lösning till behoven
- det är då systemet utformas utformas





Missbruk av User Stories

As a officer I need a menue row with the following buttons ...



Var i processen ligger verktygen

User Stories

Användningsfall

Scenarios

Funktionella krav

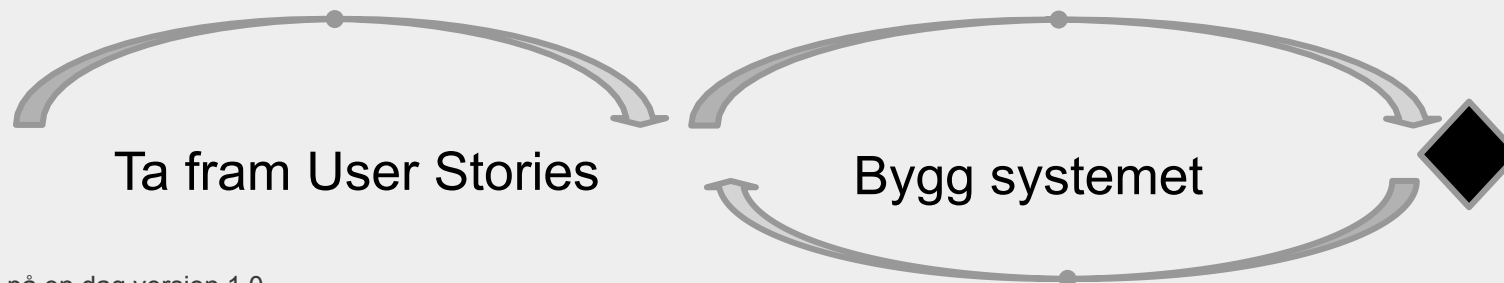
Behov Koncept Funktionella krav

Process

Scrum med äkta User Stories

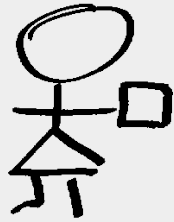


- Ser inte bergets topp
- Förtroende för utvecklarna
- Egen utveckling
- Web, spel
- Korta underhållstider





Scrum med funktionskrav



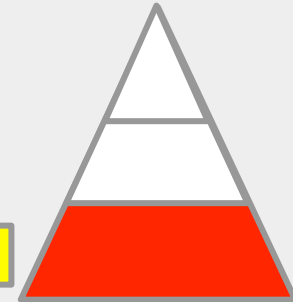
Produkt ägare



Backlog



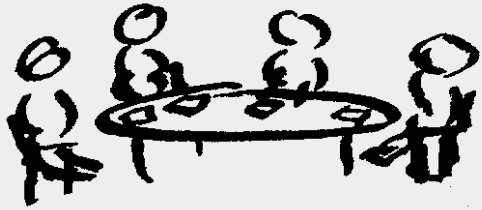
Utvecklare



- Inte ser bergets topp
- Saknar förtroende för utvecklarna
- Kontraktsutveckling
- Web, spel, adm
- Korta underhållstider



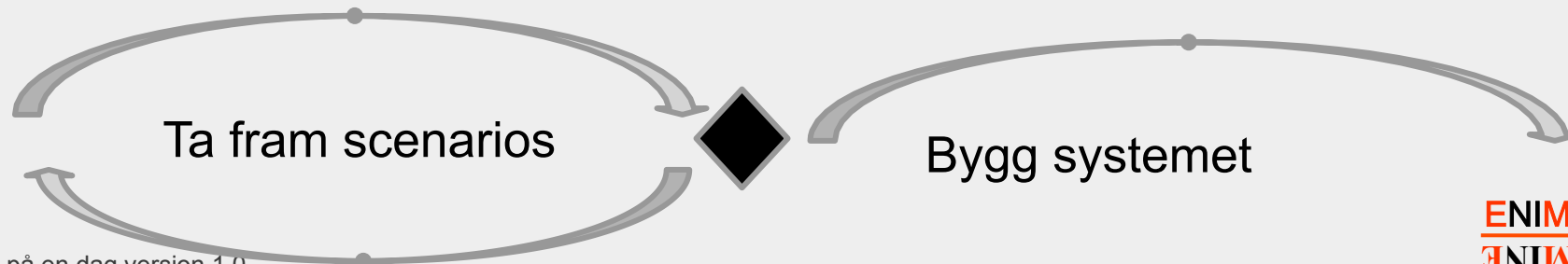
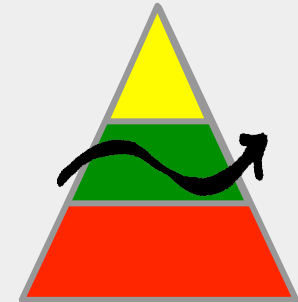
Scenariodriven development



Användare
Utvecklare
Testare



- Ser bergets topp – i viss mån
- Bygger förtroende för utvecklarna
- Egenutveckling
- Granskningsbart
- Adm, kritiska system
- Långa underhållstider

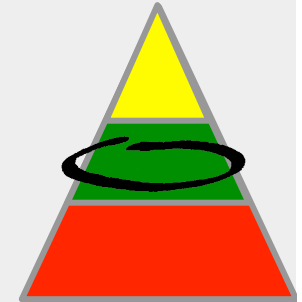


Användningsfallsdriven utveckling

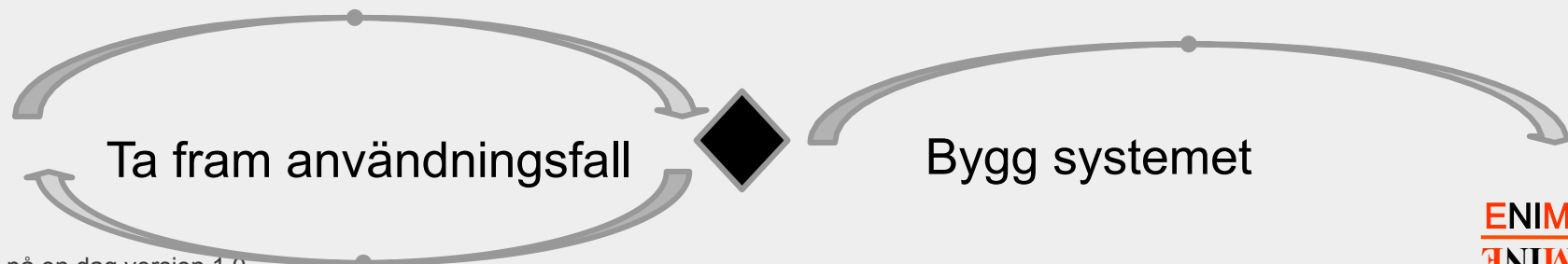
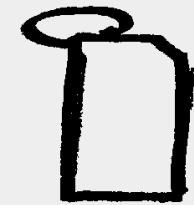


Vem skriver användningsfallen?

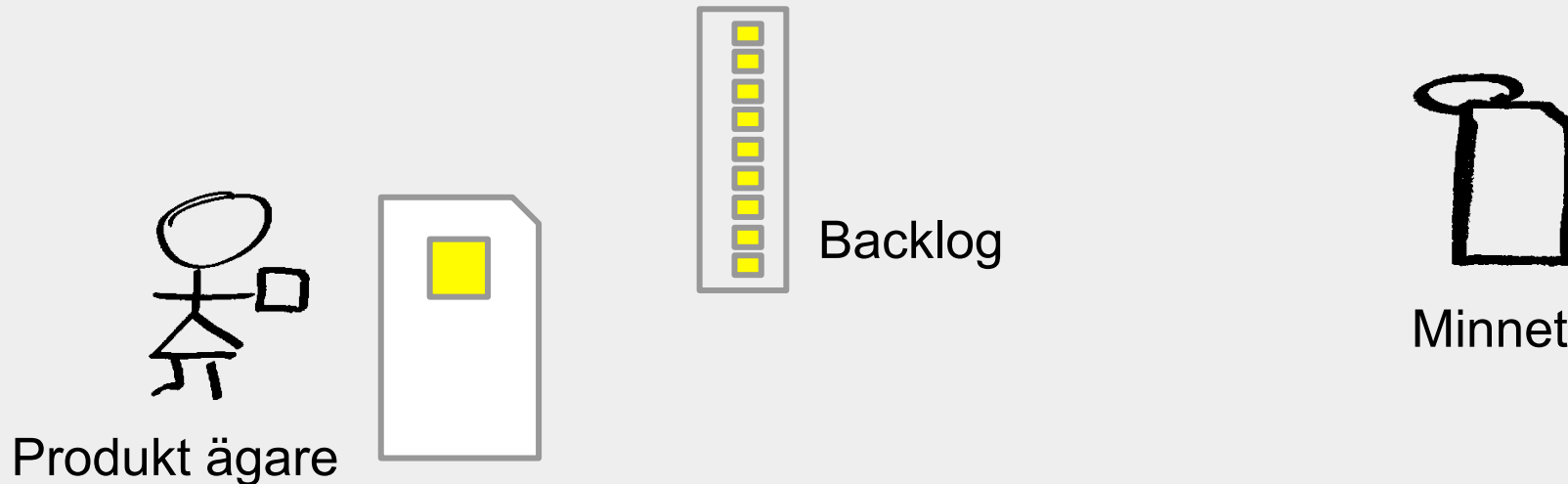
Kravare Utvecklare?
Användare? Testare?



- Ser bergets topp – i viss mån
- Bygger förtroende för utvecklarna
- Kontraktutveckling, fast pris
- Granskningsbart
- Prylar, adm, kritiska system
- Långa underhållstider

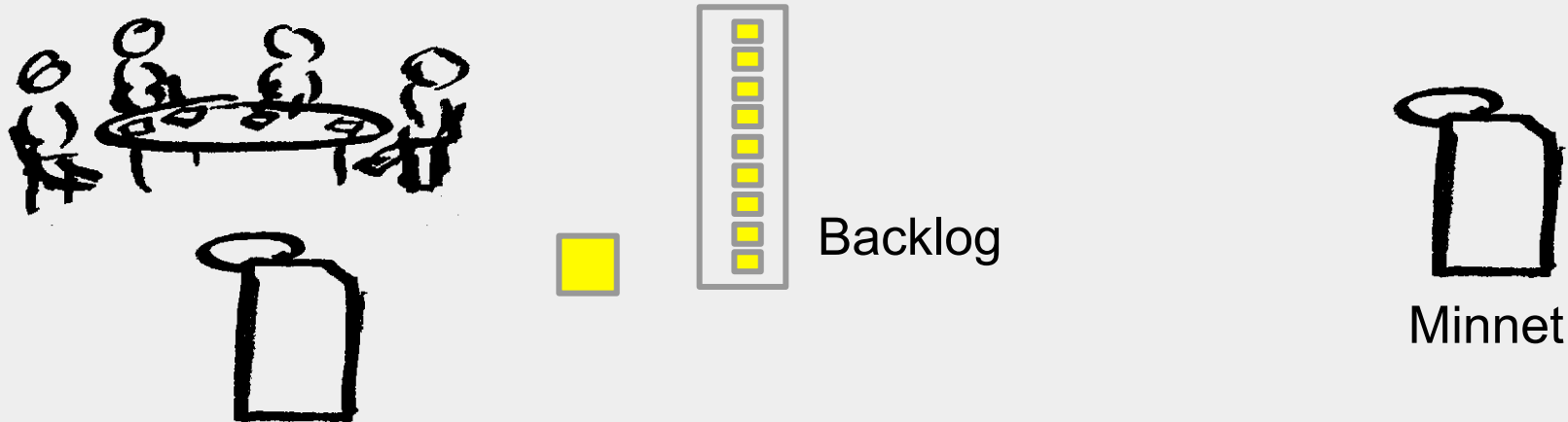


Vad ser man i praktiken?



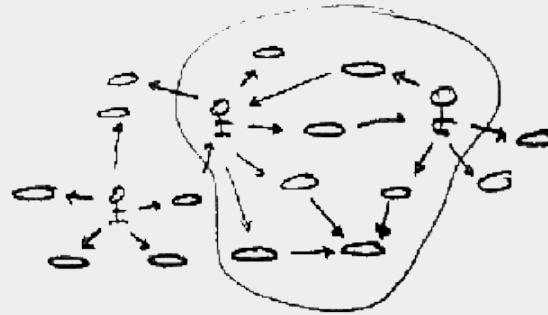
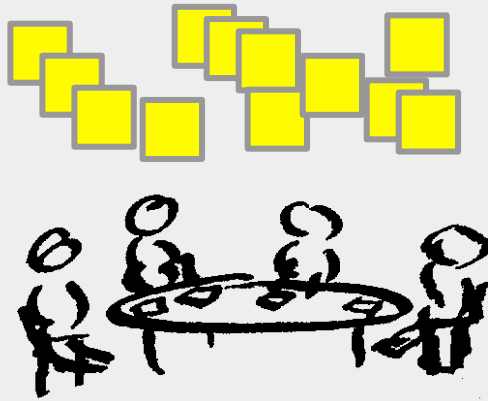
- **User Stories med testvilkor**
- **Scrum**
- **Användningsfall för minnet**

Vad ser man i praktiken?



- **Användningsfall**
- **Scrum för utveckling med funktions "User Stories"**
- **Användningsfall för minnet**

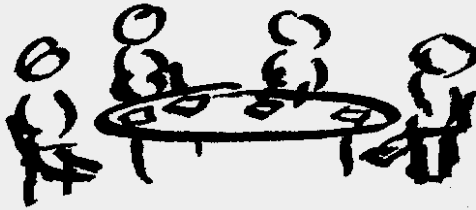
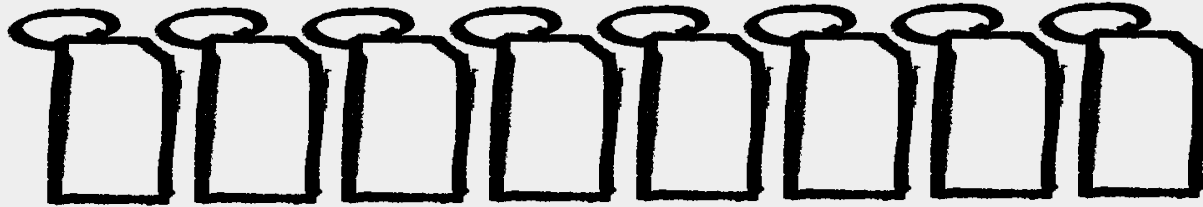
Vad ser man i praktiken?



Minnet

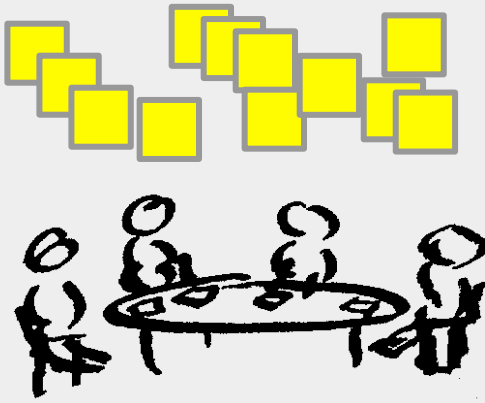
- **Kravidentifikation med User Stories**
- **Lättvikts användningsfall för att verifiera krav och test**

Vad ser man i praktiken?



- **Försök att skriva alla
Användningsfall**
- **Fast pris**

Vad ser man i praktiken?



- **Kravidentifikation med User Stories (fast de är inte äkta)**
- **Fastpris....**

Vad ser man i praktiken?



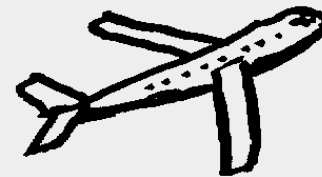
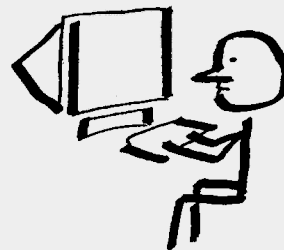
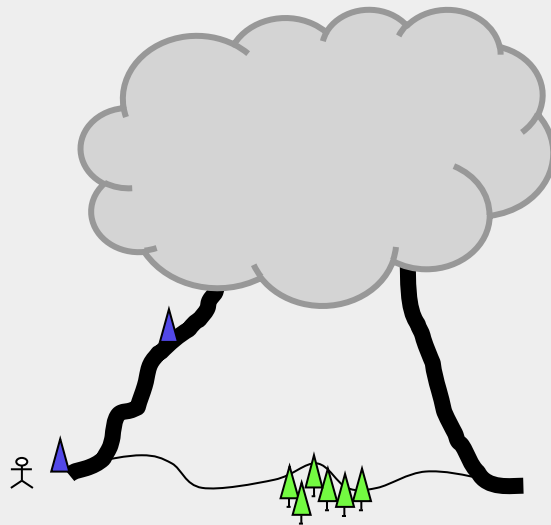
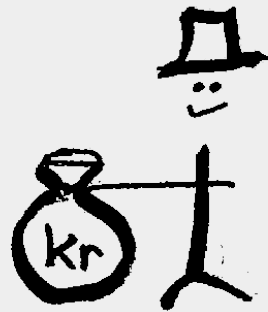
- **Rena funktionsdrivna projekt**

Vad ser man i praktiken?



- **Metod lösa**

Tänk efter först



Att förstå krav är att lägga pussel

