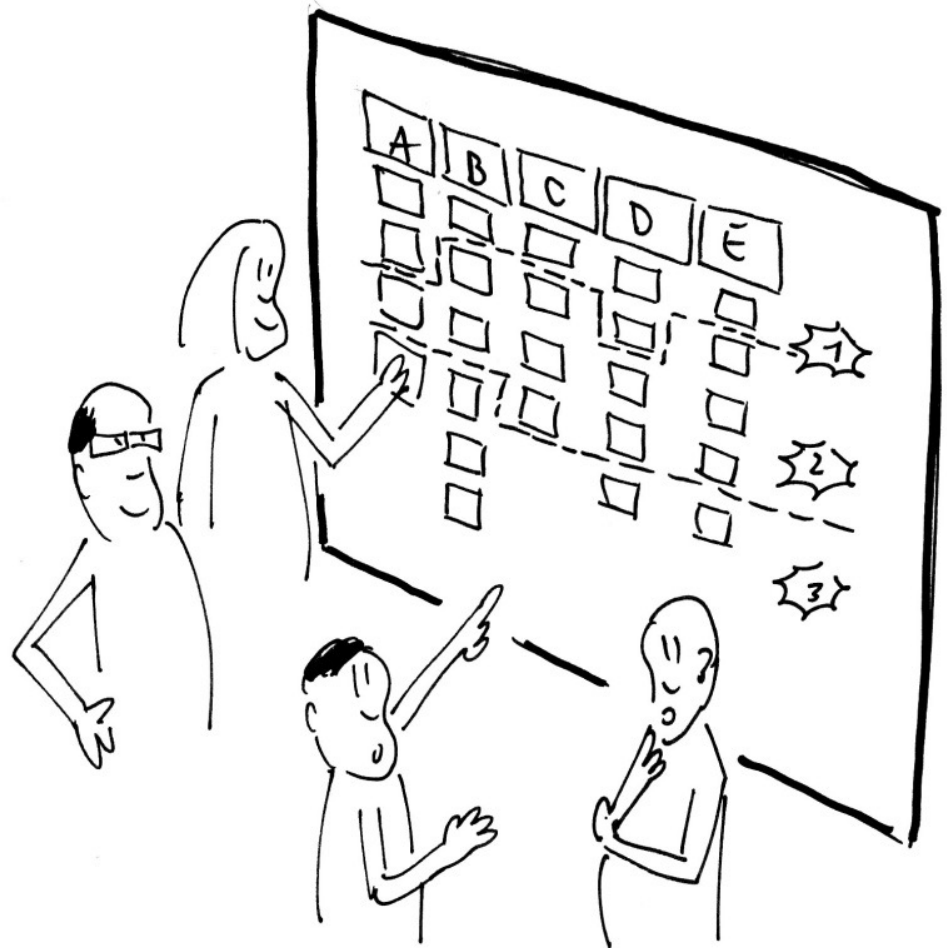


User Story Mapping

Konsten att dela upp kravbilden
på "rätt" sätt

Från Krav till System
Stockholm, 17 januari



Christophe Achouiantz – Lean/Agile Coach

@ChrisAch

<http://leanagileprojects.blogspot.se/>

User Story Mapping

Uppfanns av **Jeff Patton**,
Pask Award 2007 (USA)

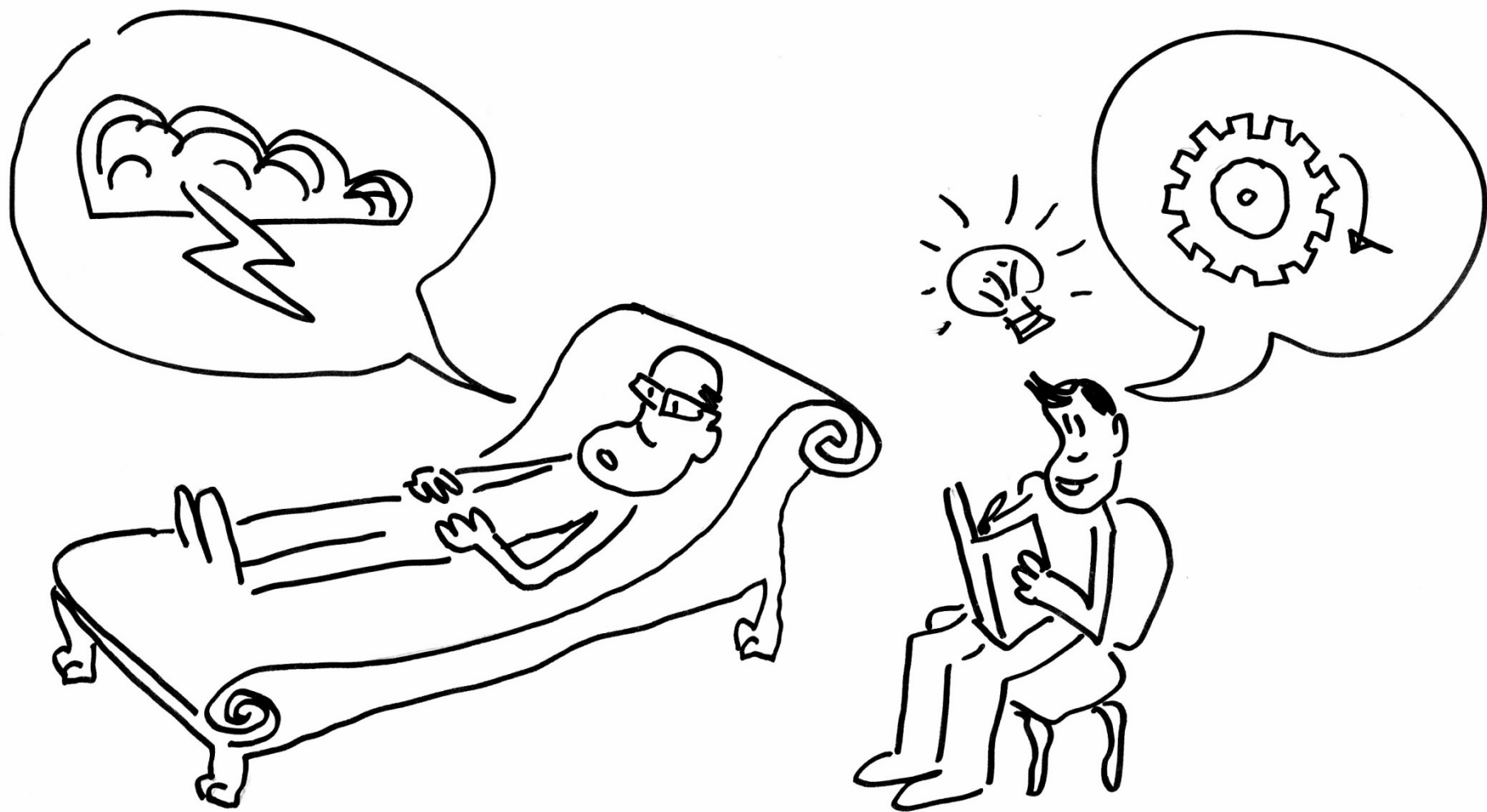
<http://www.agileproductdesign.com/>



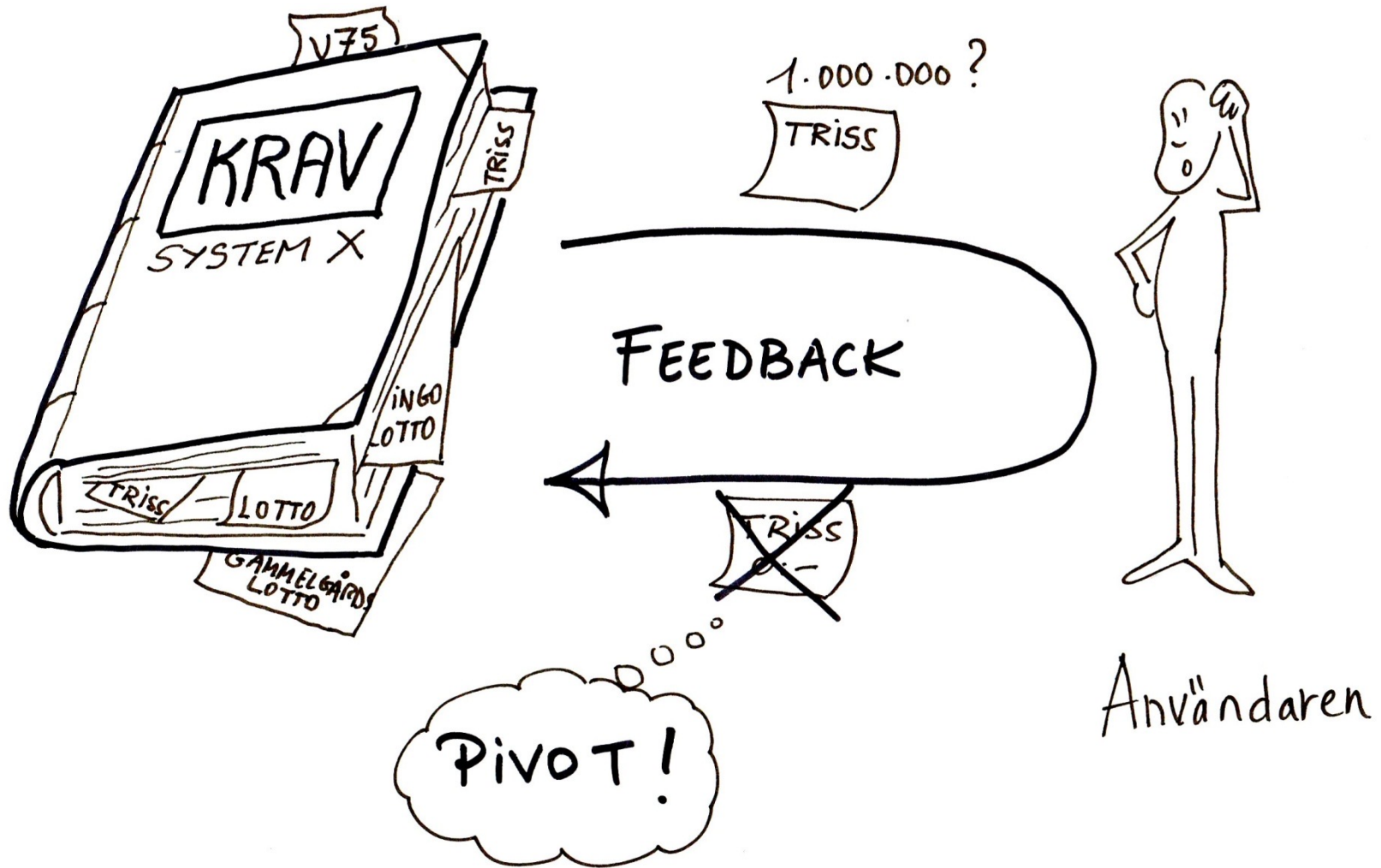
Agenda

1. **Behovet** av att dela upp en kravbild på "rätt" sätt.
2. Introduktion till User Story Mapping via **ett Exempel**.
3. **En modell** för hur man använder User Story Mapping.
4. Reflektioner.

Vi har blivit bättre på att hjälpa kunder förstå vad de **behöver**

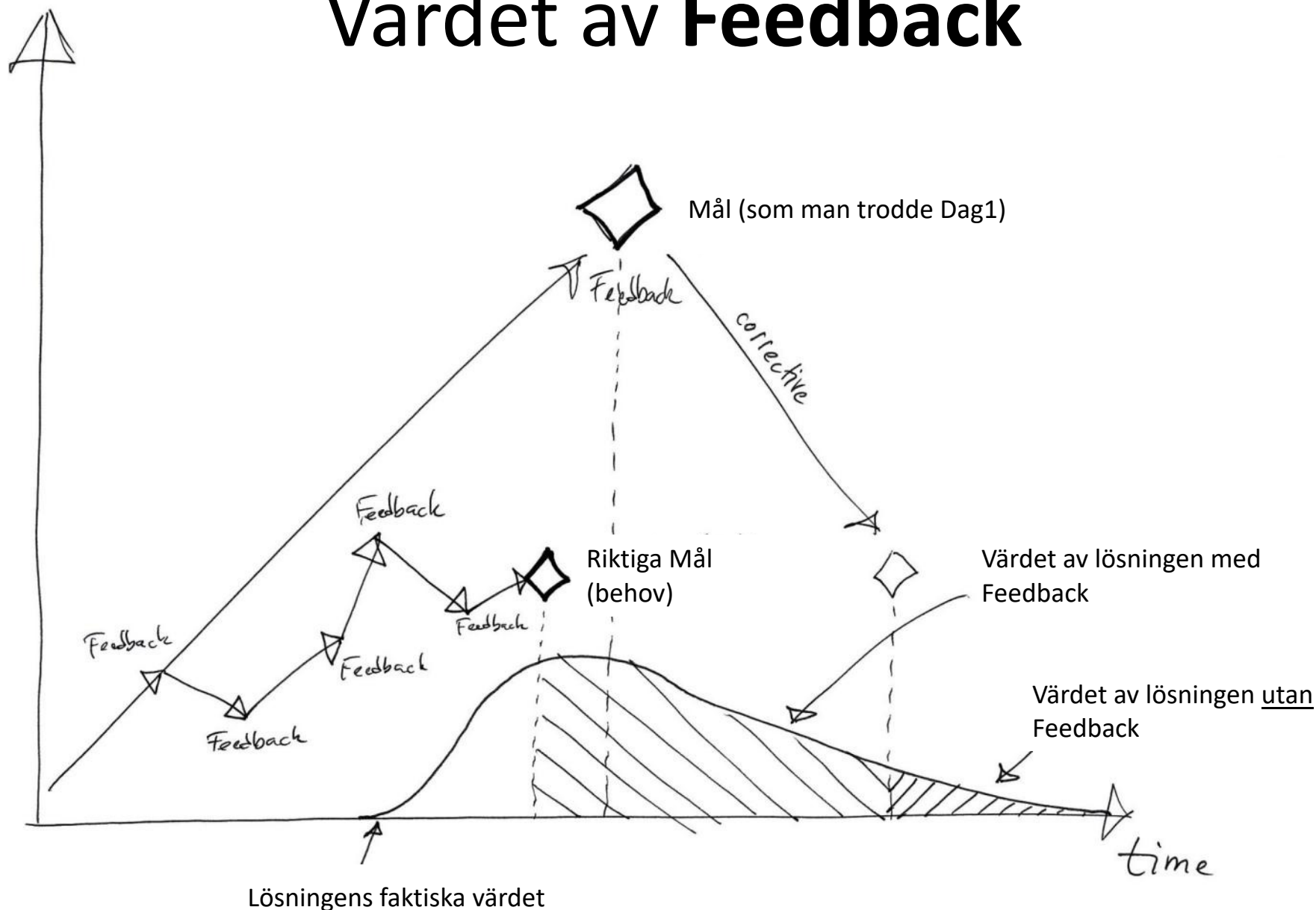


Men, kraven innehåller alltid **risker**...

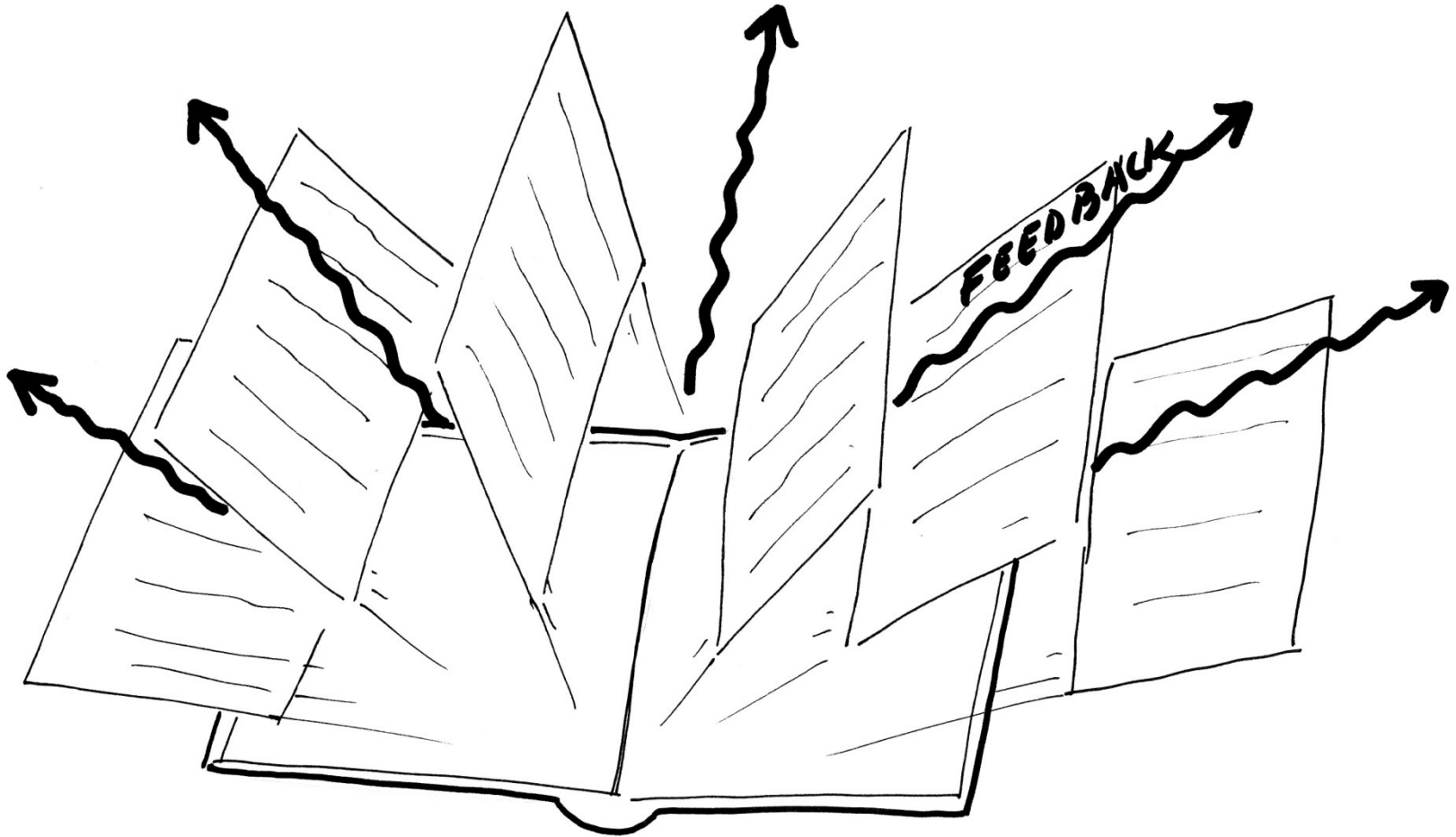


...som hanteras genom att få Feedback

Värdet av Feedback



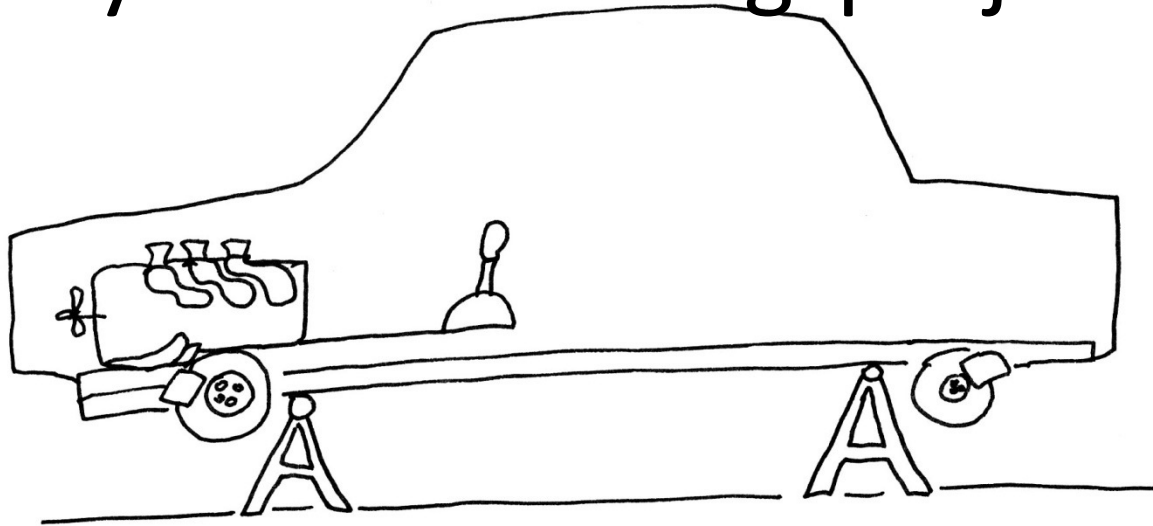
Därför, måste man **dela upp** kravbilden
baserat på den feedback man söker



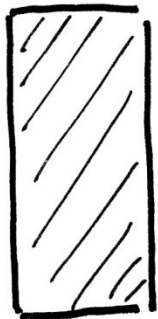
Risker i ett system



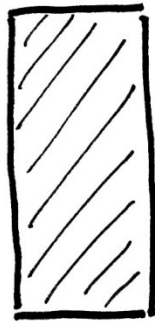
Att bygga en bil som ett systemutvecklingsprojekt



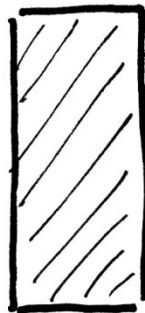
ABS
Carbon
Ceramic
superior
precision
control



BROMS



MOTOR



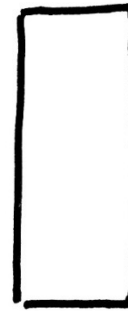
VÄXEL



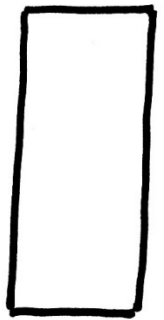
KAROSS



FJÄDRING



INTERIÖR
&
SÄT

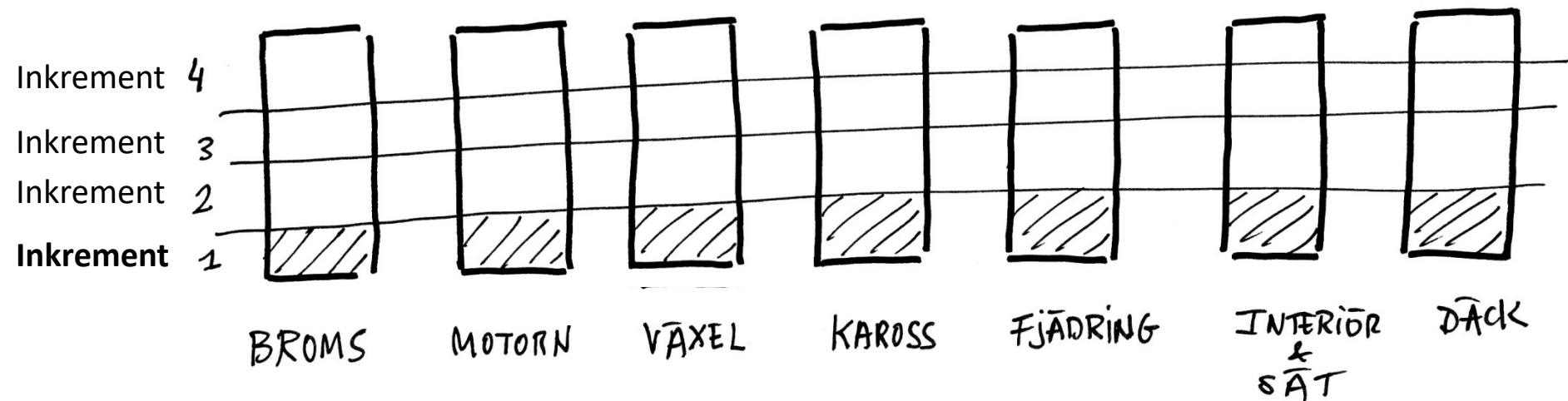
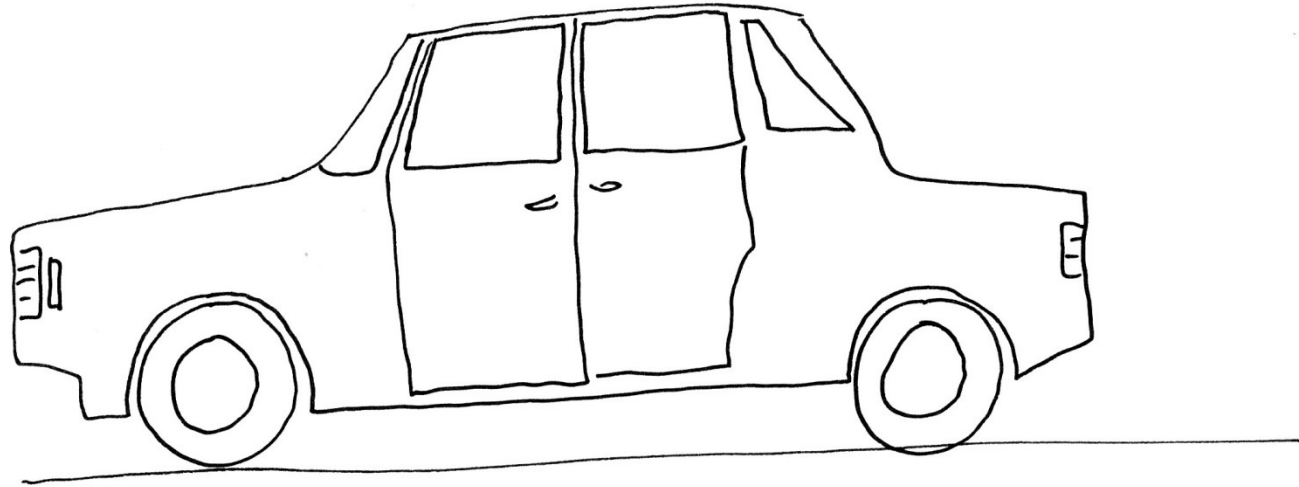


DÄCK

Stop!

Ingen tid/\$ kvar!

Att bygga en bil på rätt sätt



Sammanfattning

- Att lyckas leverera ett system på ett **tillförlitligt sätt**, och som **svarar på kundens behov**, kräver mer än bara bra krav.
- För en pålitlig leverans måste vi **minimera risker**, d.v.s. söka den **feedback** som behövs för att hantera dem.
- Detta krävs att vi **delar upp kravbilden** på rätt sätt för att få fram rätt feedback.

User Story Mapping: Ett exempel

BI Projekt

Mål

- Ett projekt för att ta fram en ny BI plattform med portal (SharePoint).
- Måste vara färdig för verksamhetskritiskt projekt PETE **om 3 månader.**

Kontext

- En **förstudie** har tagit fram en högnivådesign.
- **Inga** detaljerad specifikationer finns.
- **Oklart** exakt vad projekt PETE behöver.
- Arkitekten finns bara tillgänglig en månad innan han ska på pappaledighet.

User Story Map



PORTAL

GDW +

Kapitel
Geometrie $\leftrightarrow$ Postulate

Skapa struktur

Skapa
Rapport
innehåll

Access to
Portal

Bestall und
bill Rapp

STATISTICS

5012 UAP
Relevant ~~Data~~
♀ Report

Söka efter
Barns Namn
(eller Text Data)

NAVIGARE till
Business Roll
Function

PETE

Beroende

Validation /Tests

Frågor

User Stories

mindre "optional"

Mer "optional"



Utv. Teamet
Kunden
Användaren

Delmål /
inkrement mål

1: Structure

2: Control

3: Security

4: Data quality

5: Statistics (xtra)

6: Performance (xtra)

7: Interface (xtra)

Diskutera, klargöra, dela-upp, ändra,
validera, bekräfta

Klar for Project
"Pete"

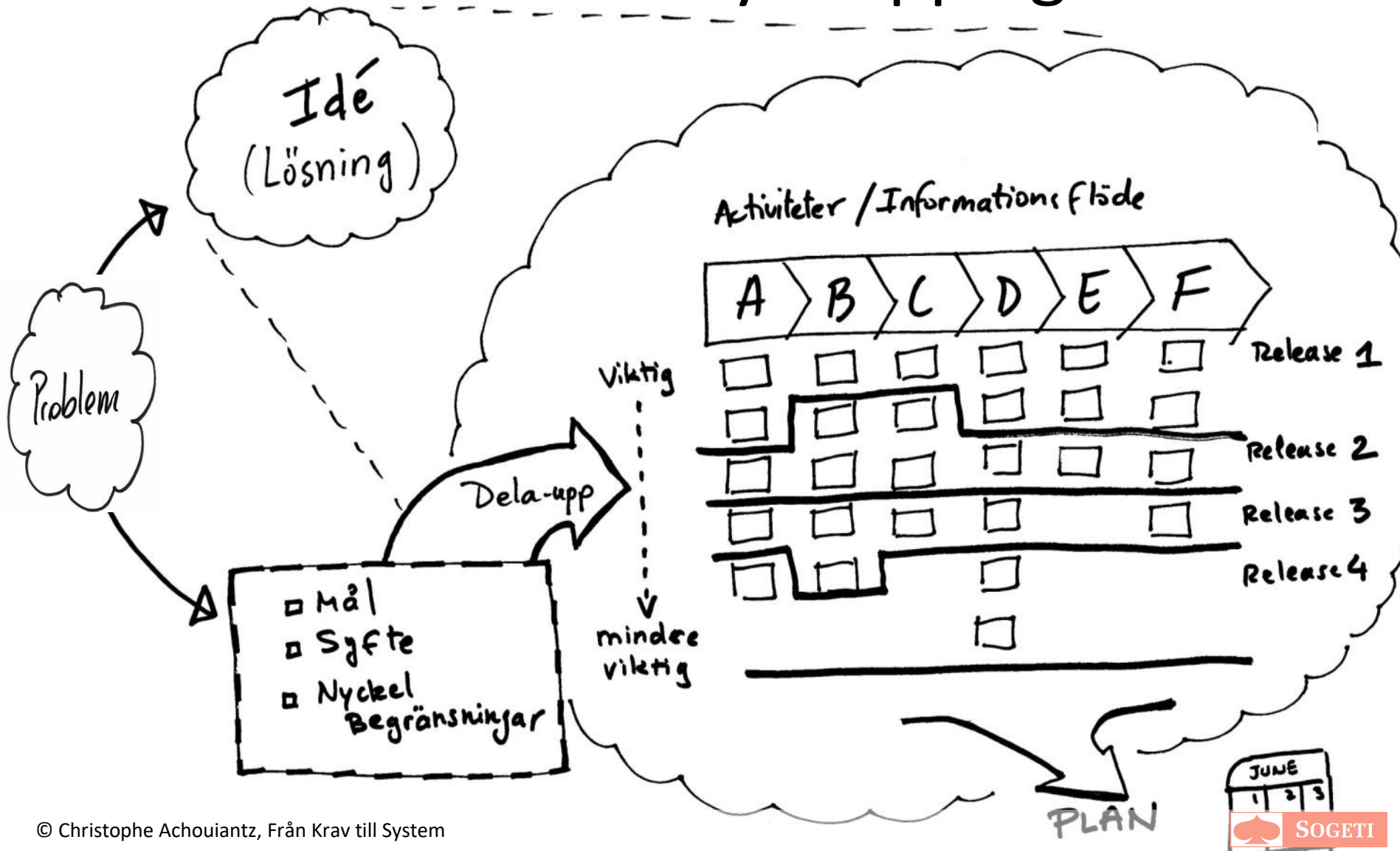
Inkrement 1:
"Basic"

Inkrement 2
"Testa
Kopplingar"

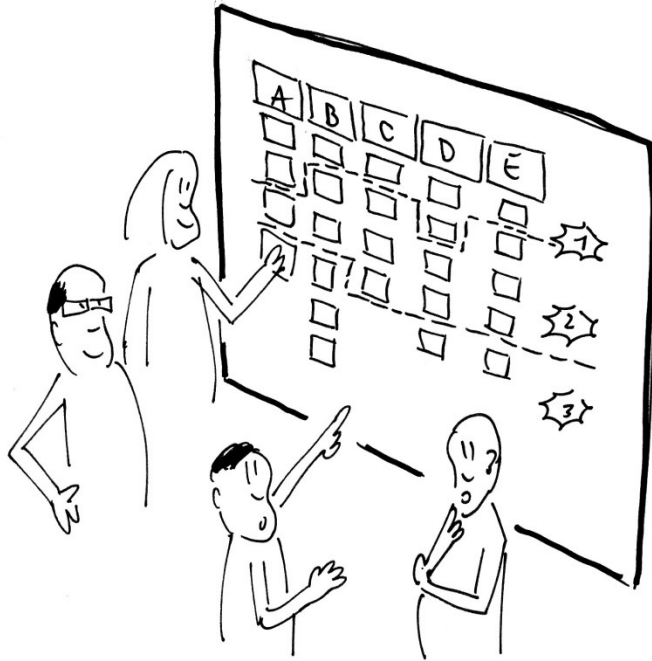
Inkrement 3
"Säkerhet"

Inkrement 4
"Kvalité"

Från idé till plan via User Story Mapping

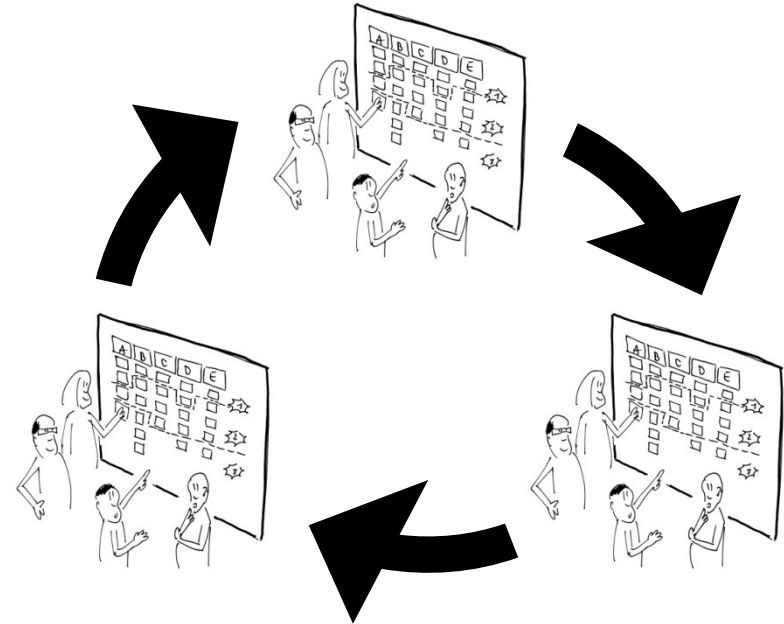


En modell för hur man använder User Story Mapping



En System Design Workshop

- Bjud in de *viktigaste/nyckelintressenter*.
- Ta fram **detaljer** kring systemet tillsammans.
"Borra" ner tillräckligt för att avtäcka och hantera de flesta problem.
- Ta fram en **visuell "kravkarta"** som stöd för diskussioner.
- Ta fram en **utvecklingsplan/roadmap** där man delar upp kravbilden i små inkrement.



Den **plan** som togs fram på workshop måste **justeras kontinuerlig**, pga:

- Nya *förståelsen* man har för systemet.
- Påverkan av *feedback* man får efter ett inkrement.
- Livet!

Ett återkommande uppföljningsmöte för omplanering

- Hantera ändringar löpande (t.ex. veckovis).
- Designa "in" ändringarna i kravkartan.
- Ändra planen efter det.

System Design Workshop: Agenda

1. Bekräfta **värdet** av systemet (Varför?).
2. Identifiera **mål och delmål**.
3. Identifiera väsentliga **begränsningarna**.
4. Identifiera **aktiviteter** i systemet.
5. Identifiera **detaljer** för varje aktivitet (story).
6. **Diskutera**, ändra, validera, bekräfta.
7. **Dela upp** kravbilden i små inkrements.

User Story Mapping

Identifiera aktiviteter

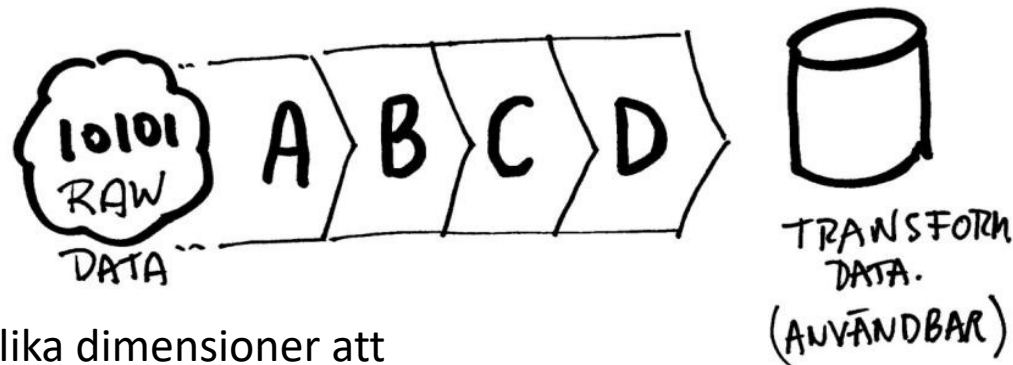
- För ett bra designarbete måste alla intressenter förstå systemets **omfattning**.
- Visualisera vad systemet gör genom en enkel aktivitetsdiagram som **visualiserar flödet av information**.

GUI: Flödet av användaraktiviteter (t.ex: login, hantera nya email, skicka email, ...).



Om det finns olika dimensioner att välja från, välj någon som de flesta intressenter är komfortabla med.

System utan GUI: Flödet av information i systemet (transformation av data).



Identifiera detaljer/Stories

- För varje aktivitet, identifiera de olika **subaktiviteter** eller uppgifter (Tasks) som ingår, gärna i form av user stories.

**Nyckeln i metoden är att alltid börja med det absolut enklaste:
*"Vad är det absolut enklaste sätt att göra/uppfylla aktiviteten?"***

- När man har identifierat det absolut enklast kan man *gradvis* öka systemets komplexitet eller rikedom.

Identifiera delmål & Inkrement

- Målet är nu att välja **en tunn skiva** av tasks/stories **som går igenom hela systemet** (aktiviteter) för att ta fram en *fungerande*, dock kanske väldigt *grov*, version av systemet.
- Senare delmål kan nu bestämmas baserat på det första delmålet. *Kunden ska helst bestämma dessa delmål.*

Varje senare inkrement måste vara:

Kort (max 1 månad),

Meningsfull (ha värde för kunden eller utvecklingsteamet),

Komplett (fungerar, helst i produktion – eller produktionslik miljö).

Några tips

- ☐ Lätt att man glömmer senare det som var "självklart" under workshopen: **fånga diskussioner** kring en "lapp".
- ☐ En testare/testledare under workshopen kan fånga **validationskriteria** för varje "lapp" och delmål – inkrementer blir konkretare och testbara (testspefikationen är i princip klar efter det).
- ☐ Bra att ha en **facilitator** under workshopen så att alla intressenter kan fokusera på systemet.
- ☐ Varje inkrement måste ha **värde** för kunden eller utvecklingsteamet.
- ☐ Bjud in så många **olika nyckelintressenter** som möjligt på workshopen.
- ☐ Försöker hålla inkrementer **max en månad långa** (helst 2-3 veckor).
- ☐ Använd "Planen" på projektmöten och med dialog med kunden så ofta som möjligt.
- ☐ Försök hålla "Planen" synlig för alla hela tiden.

Reflektioner: effekt

ENGAGEMANG
&
DELAKTIGHET
FRÅN TEAM OCH KUND

TILLIT
FRÅN KUNDEN → TEAM
TEAM → KUND
TRANSPARENT

FOKUS

FLEXIBELL
PLANEN OMPRÖVAS OCH
JUSTERAS EFTER VARJE
INKREMENT

SNABB
ATT LEVERERA FLERA SMÅ BITAR
GÅR FÖRTÄRE ÄN ATT LEVERERA
EN STOR KLUMP

Reflektioner: Varför funkar det?

DISKUSSIONER

FÖRSTÅ VALIDERA ÄNDRA
IDENTIFIERA BEKRÄFTA

VISUELLT

ALLA FÖRSTÅR HELHETEN
TYDLIGA MÅL
INTRESSENTER KAN KOMPRIMISKA

ENKELHET

FOKUS PÅ DET ENKLASTE FÖRST

ALLTID I TID

FÖRMÅGA ATT FORMA DETALJER
GÖR ATT DET GÅR ATT LEVERERA
I TID + BUDGET + KVALITÉ

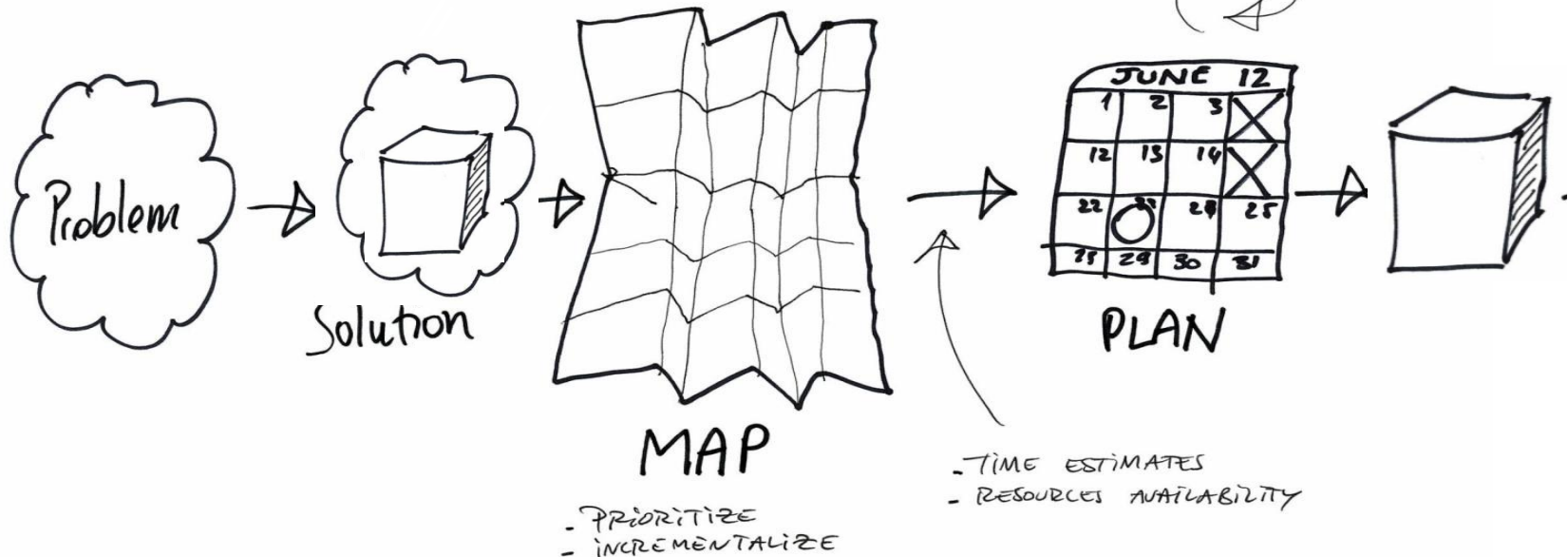
ITERATIV &

INKREMENTELL

FEEDBACK LOOPS

Sammanfattning

- Att **lyckas** med systemutveckling är att skaffa sig en **förmåga att få feedback** på det som är **riskfyllt**.
- Kravarbetet måste därför **dela upp kravbilden** för att öppna för feedback.
- **User Story Mapping** är ett *verktyg* för att göra det.



Referenser

- **Jeff Patton**
[User Story Mapping](#)
[User Story Mapping \(presentation\)](#)
[How you Slice it \(artikel\)](#)
- **Mary & Tom Poppendieck**
<http://www.poppendieck.com/>
Bok: [Leading Lean Software Development: Results Are not the Point](#)
- **Eric Ries**
[The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses](#)

Tack för er tid!

Frågor?

Christophe Achouiantz – Lean/Agile Coach

@ChrisAch

<http://leanagileprojects.blogspot.se/>

