

# L15 Att göra uppskattningar – Ansträngning och nytta

Magnus.Witting@iths.se



IT-HÖGSKOLAN

Här startar din IT-karriär.

## Incheckning

Vilket är ditt favoritplagg?



# Kort om mig



**UNDP**  
United Nations  
Development Programme



<https://www.youtube.com/watch?v=Vjq1SojOWW8>



# Dagens agenda

Syftet med dagen är att ni skall förstå svårigheten med att göra uppskattningar, att de alltid utgör ett antagande och därmed en risk. Ni skall få tekniker för att göra uppskattningar i ett agilt sammanhang

- Krav i agilt
- Feature till task
- Hur gör vi för att uppskatta något
- Tekniker för att göra uppskattningar



# Agil Kravhantering

Produktkrav (funktionella krav)  
Projektkrav (ramkrav)  
Ickefunktionella krav (Kvalitativa krav)

**Det finns krav för kraven också!**

De skall följa SMART

De kan indelas enligt MoSCoW

De skall vara accepterade av projektet **OM**  
de rymms inom

1. Budget
2. Tidplan
3. Är tekniskt möjliga
4. Har nödvändiga resurser

**OBS! ett önskemål  
är inte ett krav!**



**Skillnaden är att vi får  
regelbunden feedback och  
gillar förändringar som ger  
mer värde**

# Kano Modellen – Vad ger mest värde?

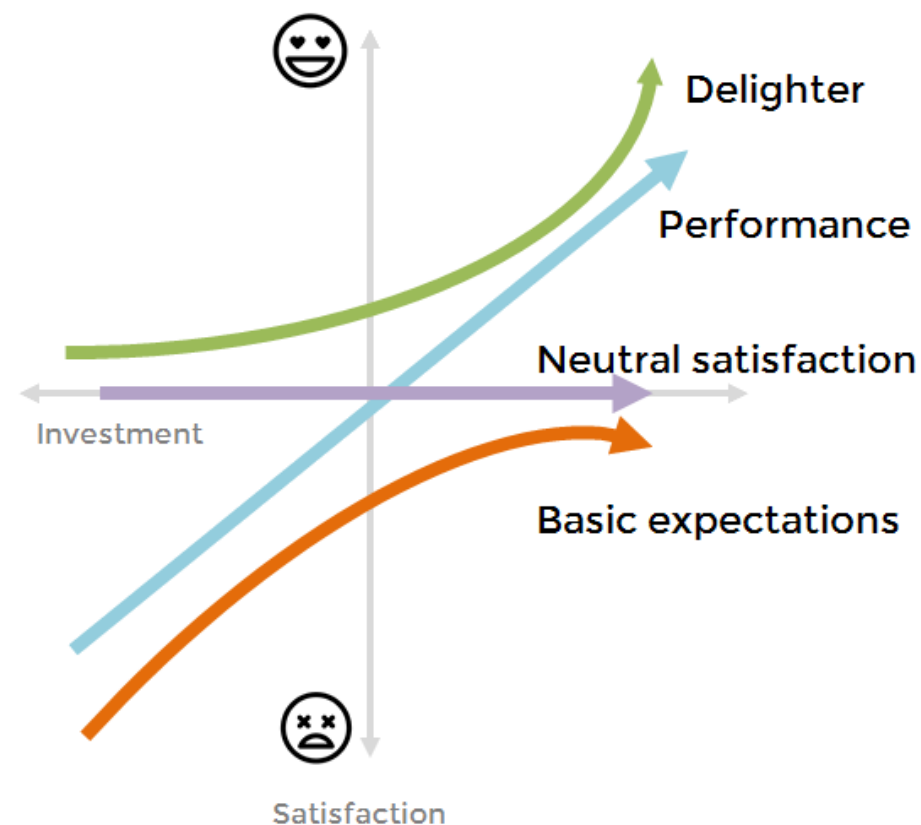
**Hygienfaktor** – Det användaren förväntar sig, ett rent hotellrum

**Endimensionell** – Användaren blir nöjd eller missnöjd, styrs av förväntningar, tid vi behöver vänta i kundservice tex.

**Attraktiva** – När vi levererar något oförväntat, varma handdukar på flyget

**Likgiltig kvalitet** – Något användaren inte uppfattar alls, en fantastisk dokumentation av lösningen

**Prestanda** – Alla användare är olika, en app med många funktioner uppskattas av vissa men andra tycker den blir svår använd



# Att bryta ner krav i flera steg

**Epic** – är en övergripande funktion. Tex ett helt steg i en e-handelssida (söka, välja produkt, betala för produkt)

**Feature** - är en delmängd av en Epic, tex att söka efter en produkt

**User Story** - är en delmängd av en feature, tex söka efter produkt med fritext

**Task** - är allt som måste utvecklas för att utveckla en specifik User Story, tex skriv en databasfråga, lägg till ett sökfält i användargränssnittet



EPIC  
FEATURE  
USER STORY  
TASK

För att söka efter en produkt behöver vi bland annat en databas och ett produktregister. Detta är features i sig själva och beskrivs inte i varje US som använder dem. De är snarare funktioner som specifika US är **beroende** av

# Vad är en User Story

- En user story är en informell beskrivning av en funktion i ett system, med hjälp av vanlig svenska.
- De skrivs ofta med utgångspunkt från slutanvändaren
- Kan dokumenteras på en post-it för att skapa överblick över hur den hänger samman med andra US och kan grupperas tillsammans för att beskriva en större funktion i systemet
- Kan skrivas av olika intressenter
- Syftet är att underlätta kommunikation mellan olika intressenter med olika kompetenser och sätt att uttrycka sig





# En komplett User-Story innehåller

Vem?  
Vill?  
Varför?

En beskrivning av vad den är beroende av...

En uppskattning av vilken ansträngning (effort), som krävs för att utveckla den

Någon form av prioritering

Tilldelad till någon ansvarig

**UC**  
Utgångsläge  
Handling  
Resultat



Har en Definition of Done (DoD)  
Dvs. den är testbar

<https://www.productplan.com/glossary/user-story/>

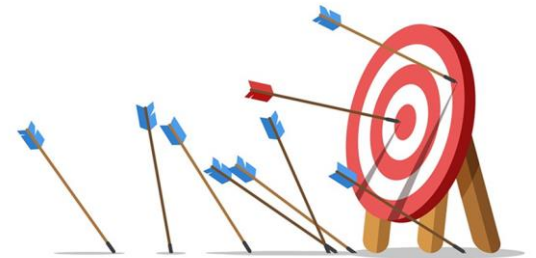
# Övning Studiegrupper tid ??

Ni jobbar med ett projekt som syftar till att erbjuda ungdomar en strukturerad och effektiv väg till att ta körkort för bil. Målet för projektet är att skapa en app där eleverna kan planera och följa upp sin väg mot körkortet

Ni skall göra en initial kravanalys för projektet:

På nästa sidan hittar ni lite blandade behov och krav efter en initial kravinsamling via ostrukturerade intervjuer.

1. Behoven och kraven är inte alltid så tydliga, så de kanske behöver klargöras/förfinas.
2. Skriv om kraven till User story format
3. Strukturera och gruppera dessa krav t ex produktkrav (funktionella krav/icke funktionella/design krav) samt ramkrav
4. Gör en första indelning av kraven i epics, features, US
5. Visa i en enkel kravspecifikation (enkel tabell)



# Krav på bilskole appen

- Eleverna ska kunna logga sina teori- och körlektioner • Det ska gå att logga både privata lektioner och lektioner i skolans regi • Man ska kunna göra noteringar och kommentarer ang lektionerna • Det vore bra att kunna ge feedback till skolan • Man måste kunna göra olika övningar och tester i appen • Det ska vara enkelt och snabbt att logga in (olika användare elever, lärare, föräldrar) • Det ska kunna gå att ändra och spara alla övningar, tester, kommentarer m m • Appen ska vara lätt att förstå sig på och enkel att hitta i – ha ett tydligt användargränssnitt • Appen bör vara snygg men inte prålig och det vore trevligt om det är kul att jobba i den • Appen ska vara klar till sista mars 2021 • Det får kosta max 500 000 sek att utveckla portalen • Den ska fungera både på plattor och mobiler • Framöver ska funktionen även finnas via webbgränssnitt • Appen ska finnas för både IOS och Android • Den ska bygga på en Cloud (moln)-lösning • Appen ska vara snabb att hantera • Både elever och skolan ska enkelt kunna se hur långt man har kommit i processen att ta körkort • Lärarna ska kunna lämna kommentarer och tips • Skolan ska kunna erbjuda lektionstider via appen • Elever ska kunna schemalägga sina lektioner • Det bör gå att via appen få hjälpa att ansöka om olika tillstånd/lämp • Det vore trevligt att kunna dela händelser i appen på sociala medier • Framöver skulle det vara bra om information från myndigheter ang körkort skulle kunnat ses/följas i appen • En första beta-version av appen ska vara klar till årskiftet 2020/2021

# User Story has Tasks

User Story

As a user I want to be able to securely log in to the application so that I can verify my user profile and change my password

Tasks

Develop the login class (1h)

Develop the scripting part for login functionality (3h)

Develop the password encryption for the login functionality (3h)

Create user table and save the login data to DB (2h)



# Att estimerera i projekt

- Allt som vi inte vet säkert om framtiden behöver vi uppskatta (estimerar, gissa, anta, beräkna...)  
Alla våra uppskattningar innehåller osäkerheter av olika storlek
- När vi skall uppskatta hur lång tid det tar att göra enkla rutinuppgifter (knyta skorna) är osäkerheten låg
- När vi skall uppskatta komplicerade uppgifter är osäkerheten högre (en längre resa med flera byten)
- När vi skall uppskatta komplexa uppgifter är osäkerheten som högst (bygga en app för att planera resor med flera byten)

En uppgift är komplex när vi inte med säkerhet kan anta att en aktivitet leder till ett visst unikt utfall – för att minska osäkerheten behöver vi skapa experiment

En uppgift som är klar för någon med erfarenhet kan vara komplex för en nybörjare

# Agila metoder är ett sätt att hantera risk

1. Osäkerhet och antaganden kan likställas med risk. (föreläsning 6)
2. Vi kan hantera risker på vanligt sätt. Om en risk är svår att uppskatta behöver vi sannolikt bryta ner den i mindre delar som tillsammans beskriver hela risken.
3. På samma sätt gör vi när vi skall uppskatta User stories. Om osäkerheten är hög behöver vi bryta isär den i fler stories



# Planering och prioritering i agila projekt

Planering i agila projekt sker på lång och kort sikt. Den längre sker genom att beställaren/produktägaren ordnar kraven i *product backlog*, viktigast överst. Planering för de närmaste veckorna gör PO/PL och teamet tillsammans och flyttar det som skall göras till en *sprint backlog*

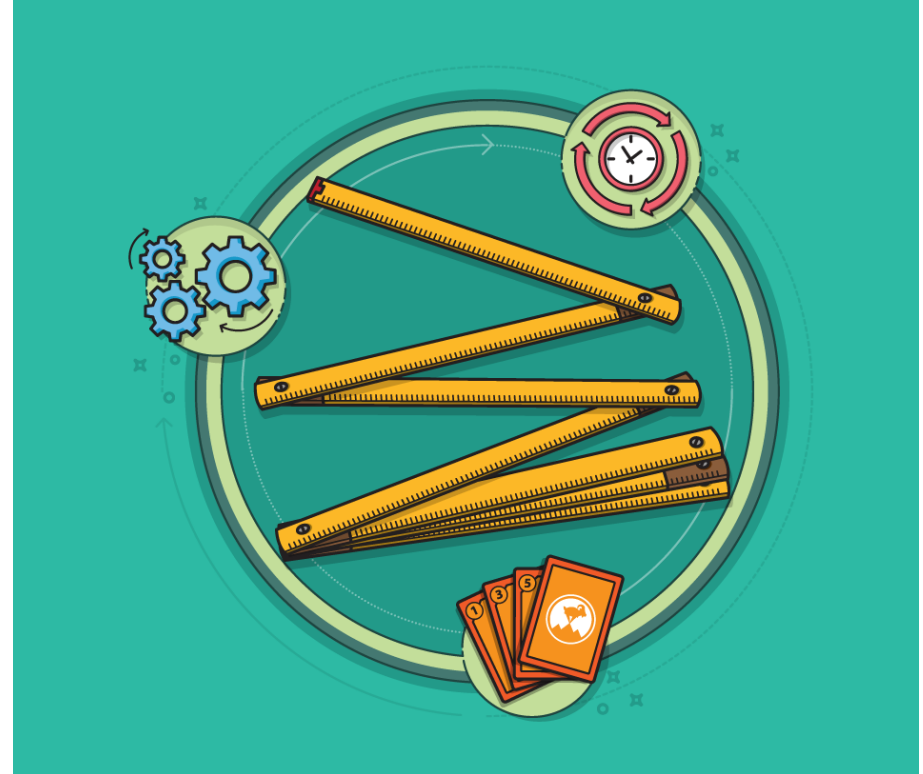
Underlaget för POs prioritering är input från användare och beställare. PO strävar efter att maximera värdet i varje sprint utan att hindra projektets övergripande värdeskapande. Underlaget för sprintplanen är en för teamet gemensam bild av **värdet** som skall skapas kopplat till den mest lämpade tekniska lösningen, som även stödjer framtida utveckling. Teamet tar också hänsyn till att allt som behövs för att kunna utveckla nästa funktion finns på plats (beroenden)

<https://www.atlassian.com/agile/project-management/estimation>

# Story Points

1. SP är ett av de mer svårbegripliga begreppen
2. Det är ett mått på ansträngning
3. Felet många gör är att likställa dem med tid
4. Ofta för att omvärlden förväntar sig att ni använder en planerad processkontroll – trots att ni jobbar agilt dvs:

**Med experimentell processkontroll**





# Story Points - Exempel

## Ett exempel på estimering för oerfarna team

**0p:** ingen komplexitet, ingen problemlösning och alla faktorer kända (typ glöm inte att)

**1p:** alla faktorer kända, låg/ingen komplexitet och kompetensen finns i teamet.

**2p:** någon del okänd alternativt att ingen i teamet löst samma problem innan

**3p:** problemet känt men information eller material saknas. (Behöver brytas ner mer)

**5p:** oklart och otydligt. Går ej att arbeta med (behöver brytas ner)

OBS! Tänk på problemen som relaterade till varandra INTE till tid eller kostnad



<https://www.scrum.org/resources/blog/practical-fibonacci-beginners-guide-relative-sizing>

# Story Points – När blir det klart?

När vi har uppskattat en US till att kräva en viss ansträngning, så har vi inte sagt något om tiden den kommer att ta

Men när vi skapar en sprintplan har vi åtagit oss inom teamet att leverera det antal SP vi planerat

För att teamet skall veta ungefär vilken ansträngning det klarar per sprint krävs övning. Med övning kan teamet göra allt säkrare uppskattningar om vilken **hastighet** (velocity) det kan hålla uthålligt

Practical Fibonacci	
0	<b>No effort is required</b> , or there is some effort required, but there is no business value delivered, so no Points are accumulated for doing the work. An example is a desired behavioral change deriving from the Scrum Retrospective.
1	<b>Extra small.</b> Developers feel they understand most requirements and consider it relatively easy, probably the smallest item in the Sprint and mostly likely completed in one day.
2	<b>Small.</b> A little bit of thought, effort, or problem-solving is required, but the Developers have done this a lot, so they have confidence in the requirements. Or, it sounds extra small, but they want to hedge their bet just a bit.
3	<b>Average.</b> Developers have done this a lot; they know what needs to be done. There may be a few extra steps, but that's it. It's doubtful that they will need to research anything.
5	<b>Large.</b> This is complex work, or the Developers don't do this very often. Most Developers will need assistance from someone else on the team. This is probably one of the largest items that can be completed within a Sprint.
8	<b>Extra Large.</b> This is going to take some time and research and probably more than one developer to complete within two weeks. In addition, Developers need to make several assumptions that increase the risk and could affect getting it Done.
13	<b>Warning!</b> This is a complex piece of work with a lot of unknowns and requires multiple assumptions to size. It is too much to complete in one Sprint. Instead, split this into multiple items that can be completed independently.
21	<b>Hazard!</b> A "21" or "34" reflects too much complexity to be done within one Sprint. It will need to be refined more. The large size also indicates more risk, assumptions, and dependencies involved to complete this item.
?	<b>Danger!</b> As a Developer, we don't want to do this work the way it is currently written. It is very complex and cannot be completed in the timeframe of an iteration or Sprint. Perhaps the

# Att estimerar tillsammans

## Planning Poker – en variant av Delphimodellen

- En facilitator
- Hela teamet ställer frågor till de är säkra på att de förstår US
- Alla skriver ner sitt estimat
- Alla visar sitt estimat samtidigt
- Om det finns avvikande höga och/eller låga värden får medlemmarna en kort stund att förklara sina värden. Sätt en tid för nästa röstning
- Nya omröstningar till konsensus råder, eller majoritetsbeslut



<https://www.scrum.org/resources/blog/practical-fibonacci-beginners-guide-relative-sizing>

# 3 Kriterier för ett bra beslut

1. Det beslut som bäst besvarar frågeställningen/når målen
2. Att alla känner sig delaktiga i beslutet och ansvar för resultatet
3. Det är tydligt vilket beslut som har fattats

Glöm inte att använda Occhams Rakkniv!

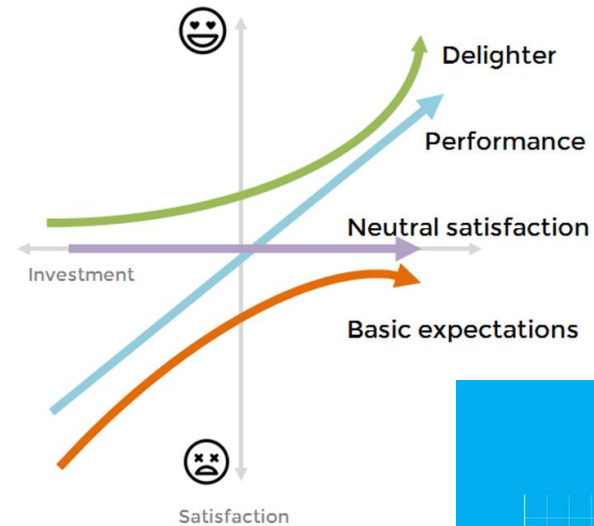




# Ansträngning vs Värde

Vi prioriterar att leverera det som ger mest värde med minst ansträngning (På epic och/eller feature nivå)

- Mest värde för beställaren
- För användarna
- För teamet (lärande, uthållig produktivitet)



# Övning Grupper om 3 tid ??

Ni jobbar med ett projekt som syftar till att erbjuda ungdomar en strukturerad och effektiv väg till att ta körkort för bil. Målet för projektet är att skapa en app där eleverna kan planera och följa upp sin väg mot körkortet

Ni skall göra en initial kravanalys för projektet:

På nästa sidan hittar ni lite blandade behov och krav efter en initial kravinsamling via ostrukturerade intervjuer.

1. Behoven och kraven är inte alltid så tydliga, så de kanske behöver klargöras/förfinas.
2. Skriv om kraven till User story format
3. Strukturera och gruppera dessa krav t ex produktkrav (funktionella krav/icke funktionella/design krav) samt ramkrav
4. Gör en första indelning av kraven i epics, features, US
5. Visa i en enkel kravspecifikation (enkel tabell)
6. Ta 10 US och bedöm hur mycket värde de tillför användaren. 1-5 poäng, använd planeringspoker
7. Försöka att uppskatta ansträngningen för att utveckla era US använd exemplet för oerfarna och planeringspoker
8. Gör en prioritering av US mer hjälp av relationen: värde minus ansträngning

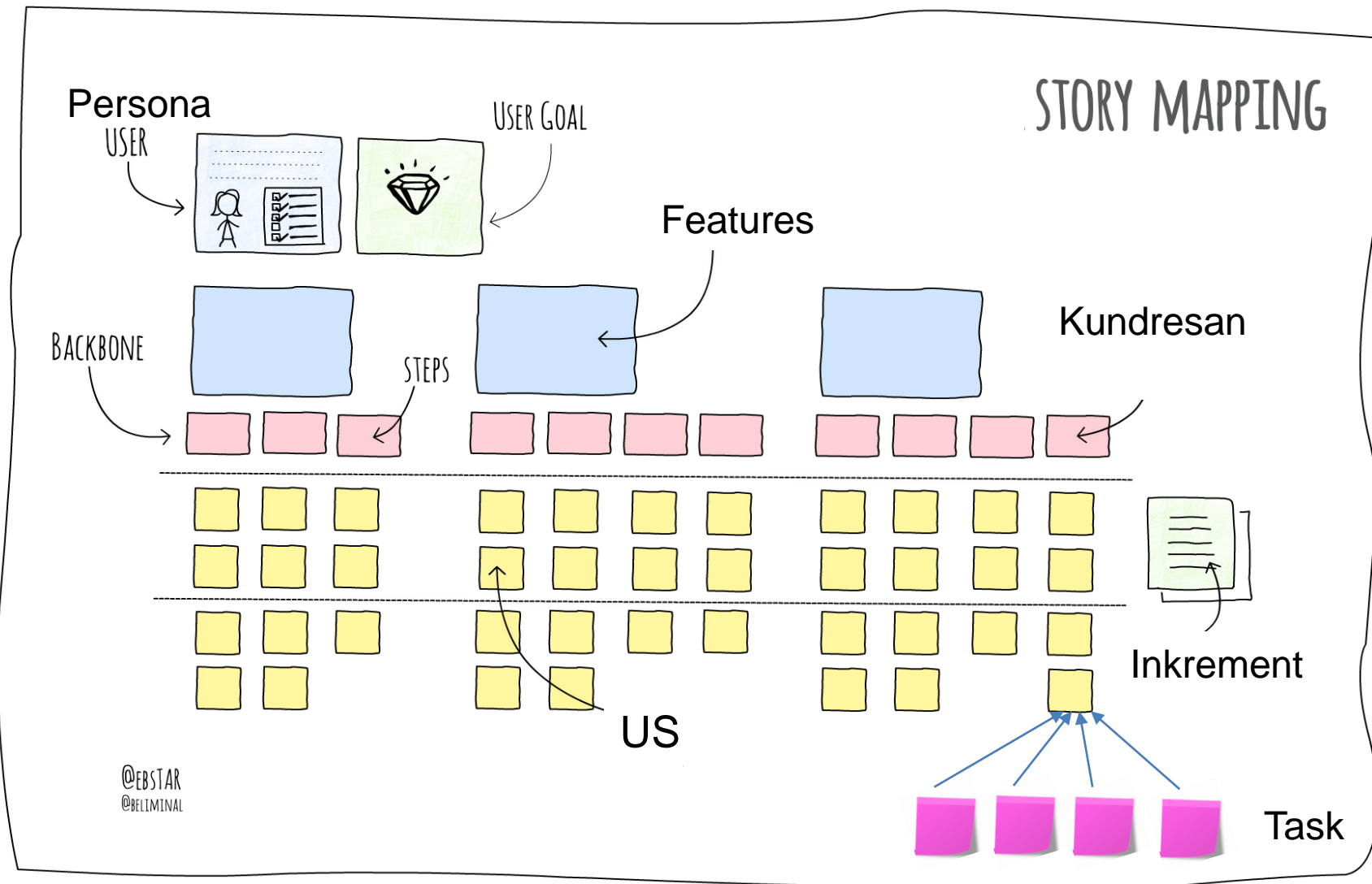


# Storymapping

- Är en metod för att visualisera produktbackloggen
- För att skapa en överblick över hur alla items passar in kundens totala upplevelse av produkten
- Horisontellt beskriver den kundresans alla steg i den ordning kunden gör resan
- Vertikalt beskriver den alla US som behövs för att förverkliga varje horisontellt steg
- När den är komplett beskriver kartan alla sätt användaren kan interagera med produkten



Ett sätt att  
visualisera  
backlogg



<https://www.jpattonassociates.com/the-new-backlog/>



# Kompetenser - Sammanfattning

Vi har koll på olika typer av krav

Vi identifierar skallkrav

Vi förstår kundens och användarnas situation, behov och vilket värde de förväntar sig

Vi förstår design och beroenden i vår tekniska lösning (teknisk kompetens)

Vi hanterar risk genom att bryta ner US i delar som minimerar osäkerhet

Vi uppskattar ansträngning och med erfarenhet kan vi bedöma vår hastighet

Vi prioriterar att leverera det som ger mest värde med minst ansträngning

# Dagens lärdomar

- Fundera på dagens lärdomar, känslor och Aha upplevelser – 1 min
- Samtala två och två - 2min
- Gemensamt 10 min



# Bokcirklar – Seminarium 2 schema

Använd din läsning och samtalen i bokcirklarna för att reflektera över: Vad har du lärt dig om dig själv, vad har du tagit med dig för lärdomar om ditt eget ledarskap och/eller den viktigaste påverkan läsningen har för hur du kommunicerar med olika medel?

Vi ägnar en timme per grupp åt att samtala om litteraturen för att ge inspiration för fortsatt läsning och träning inför det examinerande seminariet

09:00

Alex	Henryz
Casper	Henberg
Farimehr	Fazelmehr
Malgorzata	Fiedor

10:00

Atosa	Majidi
Elin	Falk
Isabella	Öberg
Maria	Wais

11:00

Anna	Magnell
Elin	Skogsberg
Feryad	Rashad
Petter	Baggeryd

13:00

Avan	Abbas
Ibuzor	Ogbowuokara
Johanna	Skans
Markus	Lermalm
Tareq	Al-samarrae

14:00

Adrian	Bradust
Bianca	Stigerud
Erik	Roos
Louise	Lindblad

# Till på torsdag 29/2

- Jag ser fram emot ett lärorikt samtal om ledarskap

**KEEP  
CALM  
THERE'S  
ALWAYS  
MAÑANA**

# Till på måndag 26/2

- På måndag kommer Markus för att rapportera vad ni eventuellt kom överens om. Förbered frågor, sen pratar vi Förändringsledning med Emelie
- Bekanta er med Azure DevOps särskilt Boards, skapa ett projekt med mallen Agile och träna. Använd <https://portal.azure.com/#home> i inloggat läge så kan ni ta självstudiekurser
- Glöm inte lämna in individuella uppgiften på söndag. Kolla skallkraven och överarbeta inte!

KEEP  
CALM  
THERE'S  
ALWAYS  
MAÑANA





**DISCLAIMER**