



Campus Nyköping

SYSTEMUTVECKLINGENS GRUNDER OCH AGIL UTVECKLING

Agil + Scrum + Kanban

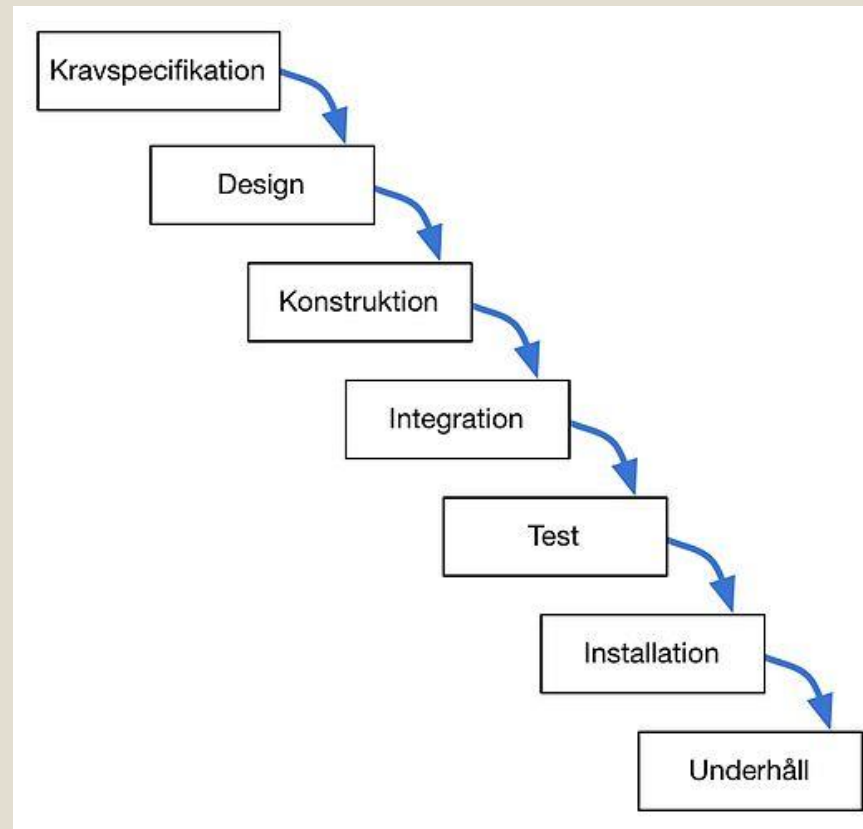
Förra gången

- Kursintroduktion
- 12 Regler för att lära sig systemutveckling och programmering
 - Kort disussionsgrupparbete med kort redovisning
- Systemutvecklingens grunder
 - Systemutveckling
 - Att arbeta som utvecklare
 - Projektmetoder

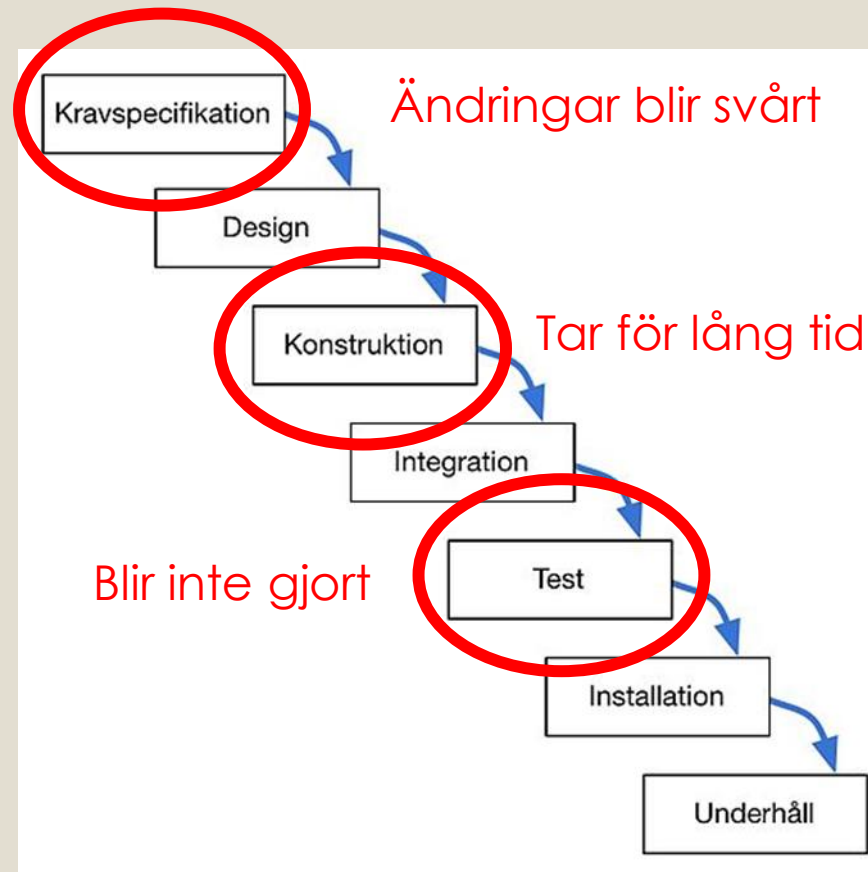
Idag

- Vattenfall
- Agila manifestet
- SCRUM
- Kanban
- User Story och Persona
- Verktyg och utvecklingsmiljö
 - Trello
- Projektarbete presenteras: "Butiken"

Innan Agilt: Vattenfallsmetoden



Problem



Vill någon ha produkten?

Vattenfallsmetoden

- Vattenfallsmodellen är en **sekventiell** systemutvecklingsprocess där man ser framstegen som ett flöde (som ett vattenfall) nedåt genom olika faser: förberedelse, etablering, analys, design, konstruktion, test, produktionssättning och underhåll.
- Modellen har sina rötter i tillverknings- och byggindustrin där det är mycket kostsamt att införa ändringar sent i processen - om inte omöjligt.
- Ett exempel som ofta används på vattenfallsmodellen brukar vara att bygga ett hus.
 - Först analyseras behoven.
 - En arkitekt anlitas som gör en ritning.
 - Denna ritning används för att ta fram specifikationer i form av olika dokument för att få söka bygglov.
 - Därefter byggs huset enligt specifikationen. Då byggnationen påbörjats är arkitekten frikopplad och inga ändringar görs.
 - Efter byggnationen sker inflyttning och drift och underhåll av fastigheten påbörjas.



Vattenfallsmetoden

- Fördelar

- **Kostnadskontroll**, beställaren (den som betalar) kan besluta i varje steg huruvida projektet ska startas, fortsätta, avslutas eller läggas på is. Ett projekt ska kunna återupptas med hjälp av de dokument som redan gjorts.
- **Resursplanering eller upphandling kan göras mellan stegen**. Om kravspecifikationen och designen är tillräckligt bra så ska vem som helst kunna implementera systemet.
- Det som levereras är **testat och är kvalitetssäkrat**.
- Vattenfallsmodellen är även **relativt överskådlig och lätt att förstå**, detta sparar tid för en projektgrupp då modellen inte behövs förklaras lika ingående i processens start.

- Nackdelar

- Vattenfallsmodellen **skjuter kvalitetsproblemen framför sig** och skapar en ökad risk för försenad leverans med fler kvalitetsproblem och **ökad kostnad**.
- Vattenfallsmodellen **hanterar egentligen inte förändringar**. Ett förändringsförslag (tilläggsbeställning) måste gå igenom flera steg för att genomföras.
- Det blir **många dokument**. Flera av dem är nödvändiga, andra dokument kanske inte kommer att läsas.
- Oftast är **datasystem mycket mer komplexa** än vad ett hus är att bygga så denna modell kan bara användas till viss del i projekt för datasystem
- Man kan inte se kundnyttan förrän i slutet av projektet
- Mer riskabelt, ofta dyrare, mindre effektivt och vanligen försenat.

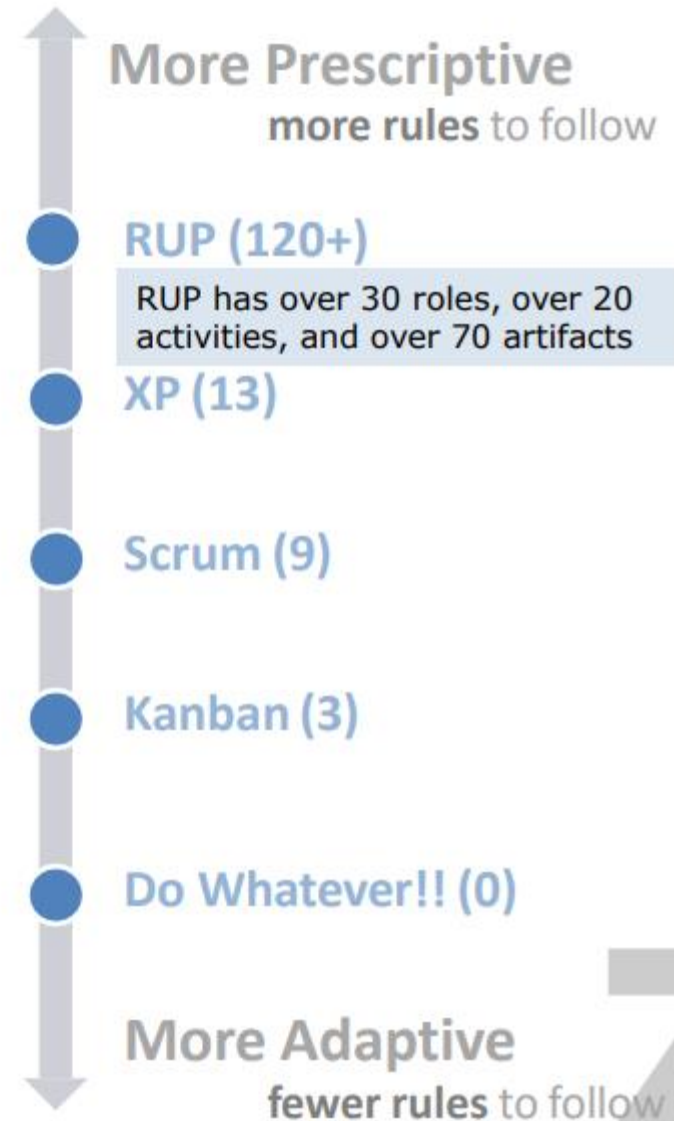
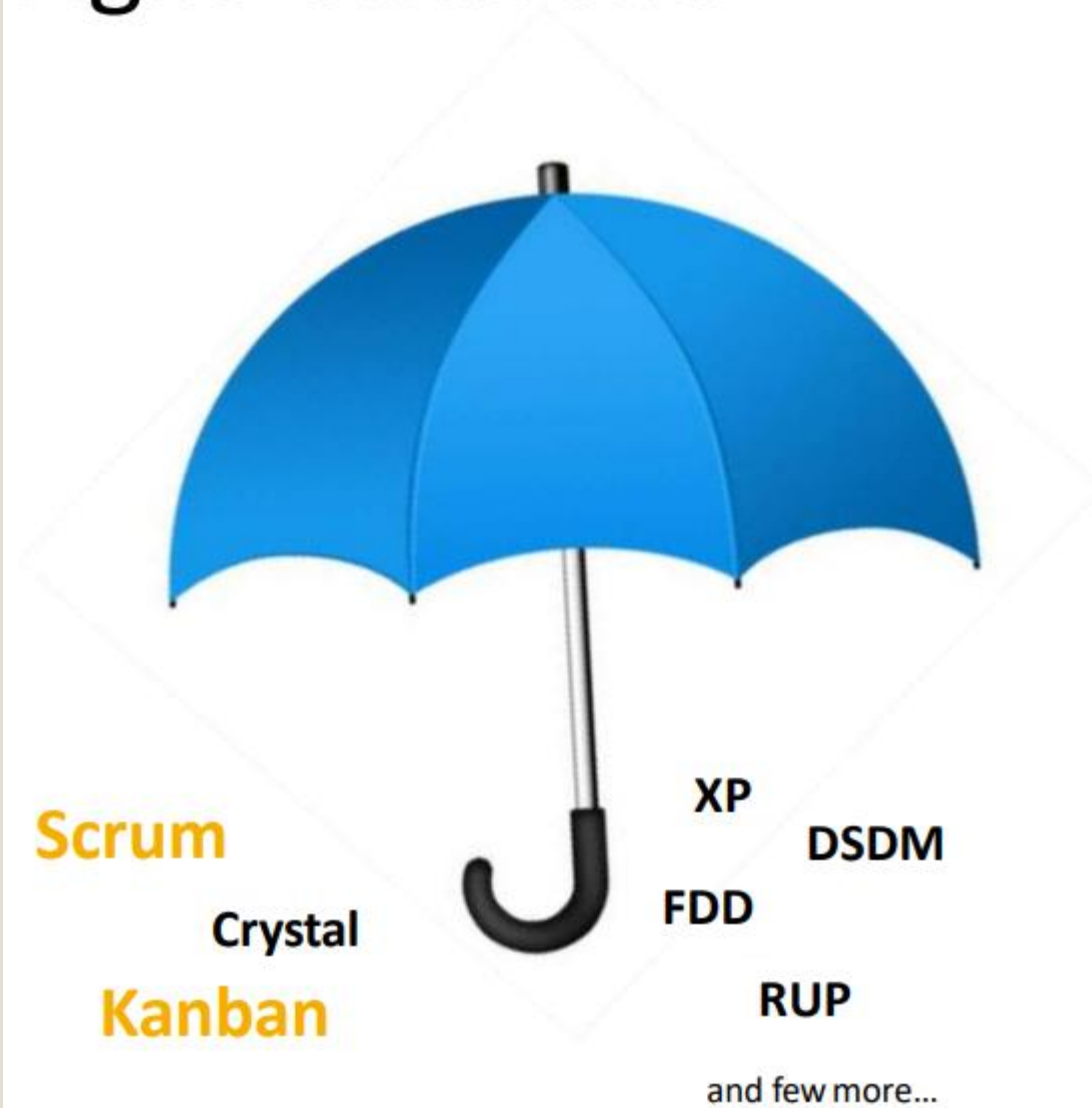
Agil utveckling

- Agil systemutveckling är ett samlingsnamn för ett antal systemutvecklingsmetoder som kan användas vid programvaruutveckling, även kallade lättrörliga metoder eller iterativa metoder.
- Det är inte en process, utan mer en filosofi eller värderingar
- Metoderna följer den filosofi och de principer som formulerades i Manifestet för agil systemutveckling 2001 av en grupp **programmerare**.
- De hade reagerat på strävan efter detaljerade kravspecifikationer, omfattande dokumentation och byråkratiserande metoder och processer som var resultatet av den traditionella projektmodellen "vattenfallsmetoden".

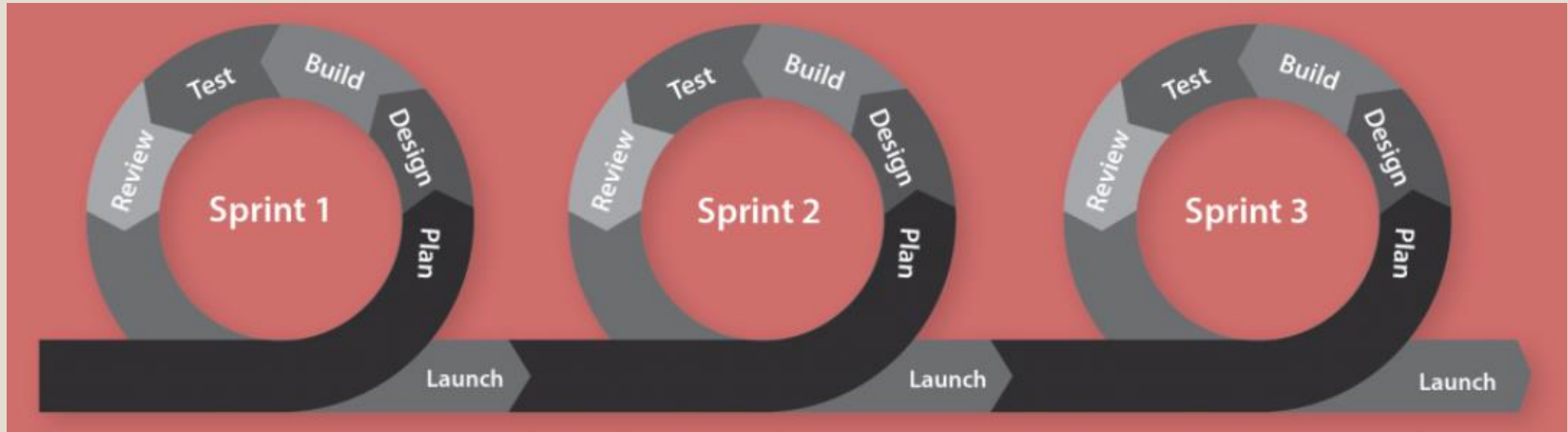
Manifestet

- Individer och interaktioner **framför** processer och verktyg
 - Fungerande programvara **framför** omfattande dokumentation
 - Kundsamarbete **framför** kontraktsförhandling
 - Anpassning till förändring **framför** att följa en plan
- Det vill säga, medan det finns värde i punkterna till höger, värdesätter vi punkterna till vänster mer.
- Individuals and interactions **over** processes and tools
 - Working software **over** comprehensive documentation
 - Customer collaboration **over** contract negotiation
 - Responding to change **over** following a plan
 - That is, while there is value in the items on the right, we value the items on the left more.

Agile Umbrella



Agil process - Iteration



Vad är agilt

- Inte en metodik
- Inte ett specifikt sätt att utveckla programvara
- Inte ett ramverk eller process
- Det är en uppsättning **värderingar** och principer
- Agilt är en tilltro eller en övertygelse som leder till beslut om hur mjukvaruutvecklingen ska ske.
- Viktigt är att förstå att agila metoder inte fattar beslut åt dig, de ger dig och ditt team grunden för att fatta egna beslut.

Principer bakom det agila manifestet

- Vår högsta prioritet är att tillfredsställa kunden genom **tidig och kontinuerlig** leverans av värdefull programvara.
- **Välkomna förändrade krav**, även sent under utvecklingen. Agila metoder utnyttjar förändring till kundens konkurrensfördel.
- **Leverera fungerande programvara ofta**, med ett par veckors till ett par månaders mellanrum, ju oftare desto bättre. Vårt mål är en veckas sprint.
- Verksamhetskunniga och utvecklare måste **arbeta tillsammans dagligen** under hela projektet.
- Bygg projekt kring **motiverade individer**. Ge dem den miljö och det stöd de behöver, och **lita på att de får jobbet gjort**.
- **Kommunikation ansikte mot ansikte** är det bästa och effektivaste sättet att förmedla information, både till och inom utvecklingsteamet. (En utmaning under pandemin)

Principer bakom det agila manifestet

- **Fungerande programvara är främsta måttet på framsteg!**
- Agila metoder **verkar för uthållighet**. Sponsorer, utvecklare och användare skall kunna hålla jämn utvecklingstakt under obegränsad tid.
- Kontinuerlig **uppmärksamhet på förstklassig teknik och bra design** stärker anpassningsförmågan.
- Enkelhet – konsten att **maximera mängden arbete som inte görs** – är grundläggande.
- Bäst arkitektur, krav och design växer fram med **självorganiserande** team.
- Med jämna mellanrum **reflekterar teamet över hur det kan bli mer effektivt** och justerar sitt beteende därefter.

Scrum

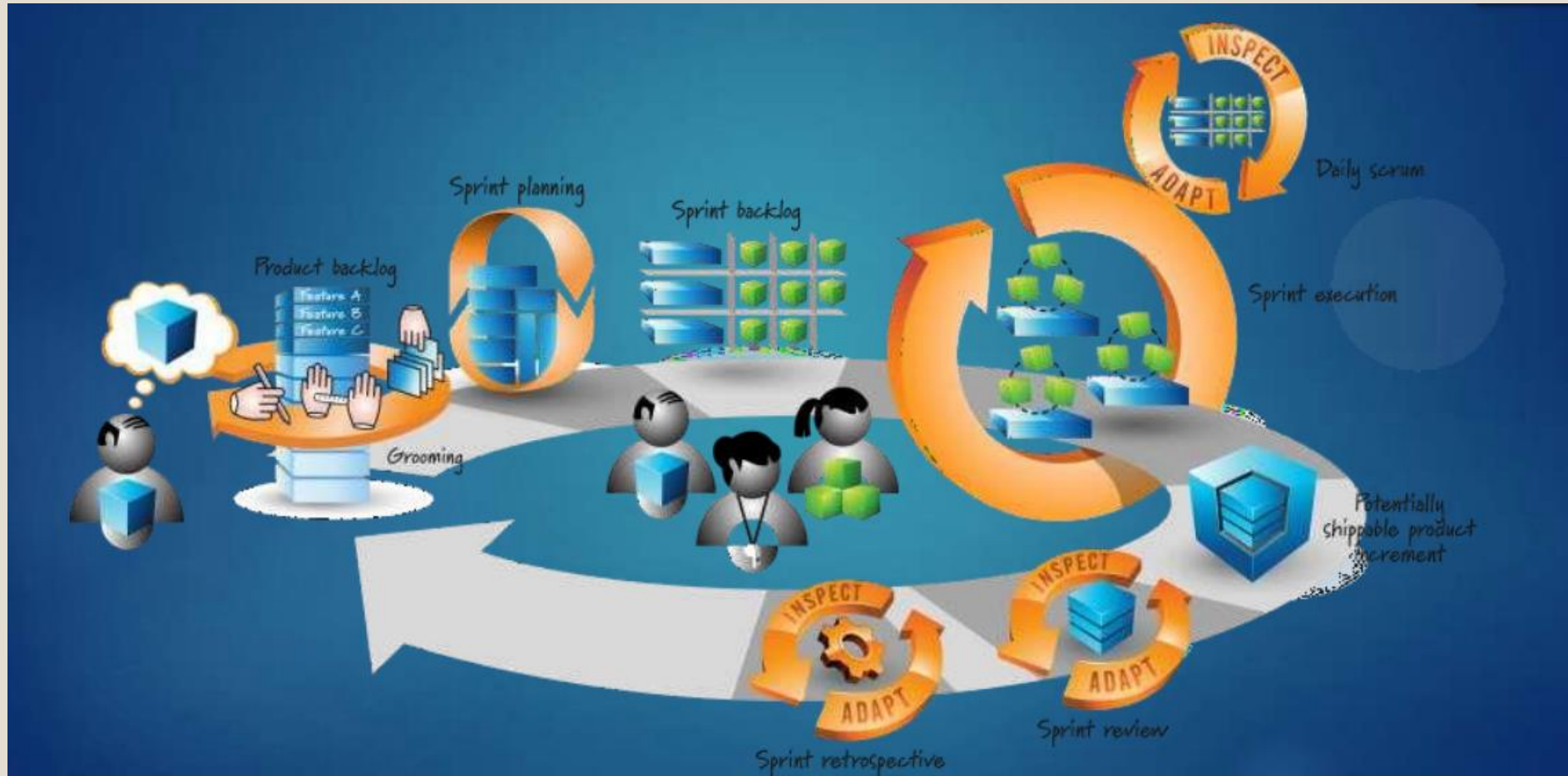


SCRUM

- Scrum är ett ramverk för att **utveckla, tillhandahålla och underhålla** komplexa produkter formulerat av Jeff Sutherland och Ken Schwaber.
- Ordet "scrum" kommer från rugby där det är ett moment när **bollen sätts i spel**.
- Scrum är ett sätt att **fördela arbetsuppgifter** i tiden med bibehållet fokus på levererad affärsnytta. Tanken är att **få ordning i projekt med kontinuerliga förändringar** i önskemålen om funktion, till exempel att kunden upptäcker ett problem under provning av en prototyp eller kommer på ett nytt bättre sätt att lösa problemet.
- Scrum ses av många som en **användarcentrerad** systemutvecklingsprocess som har **fokus på nyttan för de människor som använder systemen**.



SCRUM Framework



Projektets roller

- **Utvecklare**
- Produktägare
- Scrum master



Utvecklaren

- Att ta ansvar för att tidsuppskatta, planera, och hantera alla sina egna uppgifter och rapportera om framsteg.
- Samarbeta nära med alla andra medlemmar i teamet och att ta delat ansvar för de övergripande insatser som teamet har åtagit sig.
- Att ta ansvar för kvaliteten på den programvara som utvecklaren producerar.
- Interagerar med användare efter behov för att förtydliga krav.
- Förstå verksamhetens syfte med storyn och definiera och analysera möjliga alternativa sätt att tillfredsställa syftet.
- Arbeta direkt med Produktägaren för att klargöra och ytterligare definiera detaljerna i hur storyn ska implementeras

Produktägaren

- Produktägaren ser till att de enskilda elementen i backloggen är ordnade i rätt ordning, optimalt för projektet.
- Hen ser till att varje element i backloggen beskrivs på ett tydligt och begripligt sätt för hela teamet.
- Hen optimerar värdet av arbetet som utförs av det team som ansvarar för utveckling
- Produktägaren garanterar att backloggen skapas på ett tydligt och transparent sätt: den anger också vad Scrum-teamet har att göra i nästa steg.
- Hen ser till att Utvecklingsteamet förstår alla element som finns i backloggen till en nivå som gör det möjligt för hela gruppen att arbeta.
- I vår uppgift kommer ni alla tillsammans agera Produktägare, då ni skapar backloggen och under varje sprintplanering.
 - Därefter övergår ni till att vara Utvecklare

Scrum master

- Scrum-mästaren är den grupproll som ansvarar för att teamet praktiserar agila värderingar och principer och följer de processer och metoder som teamet kom överens om att de skulle använda.
- Ansvaret för denna roll omfattar:
 - Rensa hinder
 - Att etablera en miljö där teamet kan vara effektivt
 - Har koll på teamdynamiken
 - Säkerställa en god relation mellan teamet och produktägare samt andra utanför teamet
 - Skydda teamet från avbrott utifrån, och distraktioner.
- I uppgiften får alla ansvar för att bevaka detta, och påminna varandra om ni inte känner att arbetet sköts enligt agila metoder.

SCRUM - Beståndsdelar

- Product backlog
 - En att-göra-lista. Här samlas alla önskemål som produktägaren vill ha.
- Sprintplanering
 - Planera vilka userstories vi ska lägga in i den här sprinten
- Sprint backlog
 - En lista över det som förväntas hinnas med under **en** sprint
- Sprint
 - Arbetet delas in i sprintar. Varje sprint är mellan 3 och 30 dagar lång.
- Implementering
 - Själva utvecklingsarbetet
- Inkrementet
 - Vad vi åstadkommit under sprinten av ny funktionalitet. Det ökade affärsvärdet.
- Daily Scrum eller standup
 - Ett kort planeringsmöte för utvecklingsteamet. Det får ta maximalt 15 minuter.
 - Tre frågor: Vad gjorde jag igår? Vad ska jag åstadkomma idag? Finns det något som hindrar mig?
- Sprint Review
 - En presentation av vad vi åstadkommit under sprinten
- Sprint Retrospective
 - Teamets egna möte, där vi funderar av vad vi gjort bra, vad som gått dåligt.



Produktbacklog

- Produktbacklog är funktioner/UserStories som krävs för att uppfylla produktägarens vision och prioriteras enligt affärsvärde.
- Under pågående produktutveckling kan den också innehålla nya funktioner, ändringar av befintliga funktioner, defekter, tekniska förbättringar och så vidare.
 - Kom ihåg att det är tillåtet att även lägga in andra typer av uppgifter än userstories
- Poster kan läggas till, raderas och revideras av produktägaren när affärsförhållandena förändras, eller när Scrum-teamets förståelse för produkten växer.
- I vår uppgift ska ni tillsammans skapa och rangordna alla Userstories



Sprintplanering

- Sprint Planning Development-teamet granskar produktbackloggen och bestämmer de högst prioriterade artiklarna som teamet realistiskt kan uppnå i den kommande sprinten medan de arbetar i en hållbar takt.
- Många team delar upp varje funktion i en uppsättning uppgifter/Tasks för att bilda en andra backlog som kallas sprint eftersläpning. Utvecklingsteamet ger sedan en tidsuppskattning (vanligtvis i timmar) vad som krävs för att slutföra varje uppgift
- I vår uppgift blir det i första sprinten lite gissning, och nästa vecka pratar vi om hur vi kan estimerar arbete på annat sätt.



Sprint

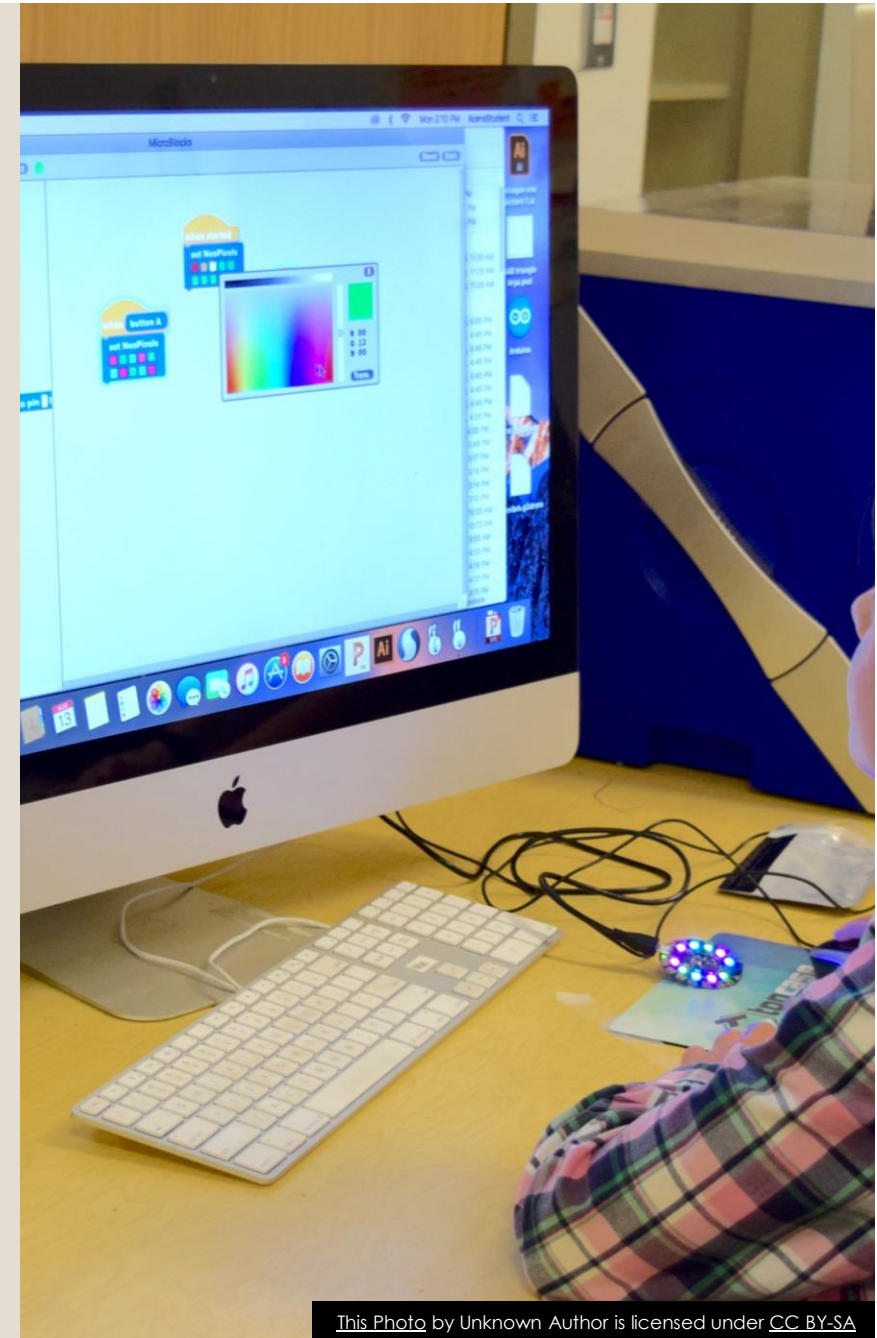
- Arbetet utförs i iterationer eller cykler på upp till en kalendermånad som kallas sprintar.
- Kalenderplanerat så att de alltid har ett fast start- och slutdatum, och i allmänhet bör de alla ha samma varaktighet. En ny sprint följer omedelbart direkt efter tidigare sprinten.
- I vår uppgift är varje sprint 3-4 dagar, och vi kommer hinna med 2 stycken.

Bättre sprint-bild



Implementation – Själva arbetet

- När sprintplaneringen är klar utför utvecklingsteamet allt arbete på uppgiftsnivå som krävs för att få funktionerna "klara".
- Ingen berättar för utvecklingsteamet i vilken ordning eller hur man gör arbetet på uppgiftsnivå i sprintbackloggen.
- Teammedlemmar definierar sitt eget arbete på uppgiftsnivå och organiserar sig sedan på vilket sätt de känner är bäst för att uppnå sprintmålet.
- I vår uppgift sker detta i grupp, där ni tillsammans genomför varje uppgift.
 - Valfritt är att dela upp arbetet på mindre grupper, men ingen bör jobba helt själv.
- Använd alltid Kanbantavlan aktivt och flytta varje User Story/uppgift till rätt kolumn.



Daily Scrum/Standup

- Daglig scrum/standup används som inspektion, synkronisering och adaptiv daglig planeringsaktivitet som hjälper ett team att göra sitt jobb bättre. Alla förstår helheten av vad som händer, hur de går mot sprintmålet, eventuella ändringar de vill göra i sina planer för den kommande dagens arbete och vilka frågor som behöver hanteras.
- Inte en problemlösningsaktivitet och inte ett statusmöte
- Max 15 minuter
- Varje person ska svara på de här frågorna:
 - Vad gjorde jag igår?
 - Vad tänker jag göra idag?
 - Finns det några hinder för mitt arbete?



Definition of Done

- När det som Scrum-teamet bestämde har blivit verklighet anses den vara "klart".
- Det slutförda arbetet ska vara av god kvalitet och kan levereras
- För en userstory så ska den tilltänkta användaren kunna göra det som bestämts, så den kan uppnå det han/hon vill.
- Userstoryn har blivit sann!





Sprint Review

- Målet med denna aktivitet är att inspektera och anpassa den produkt som byggs.
- Reviewen fokuserar på att granska just de funktionerna som utförts under sprinten.
- Alla närvarande får tydlig insyn i vad som händer och har möjlighet att hjälpa till att påverka den kommande utvecklingen för att säkerställa att den mest affärsmässiga lösningen skapas.
- Detta görs vanligen som en demo där alla är med och får se vad som byggs.
- I vår uppgift är det tänkt att vi gör detta nästa gång på onsdag morgon.
 - Kort "redovisning" av vad ni gjort. Nämn inte saker ni tänker bygga senare, eller saker som inte är klara.

Sprint Retrospective

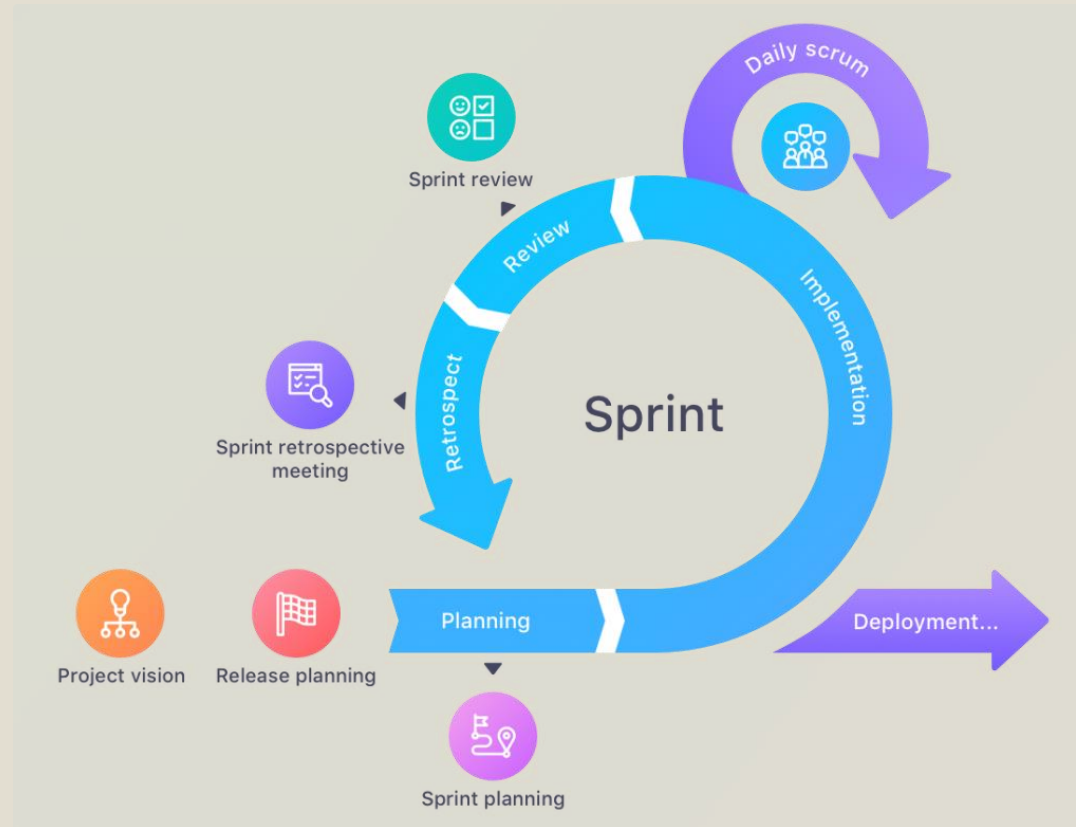
- Ett möte inom teamet där ni har möjlighet att utvärdera och anpassa processen. Utvecklingsteamet, ScrumMaster och produktägaren träffas för att diskutera vad som fungerat bra och inte fungerat lika bra, med Scrum och tillhörande tekniska metoder i fokus.
- Scrum-teamet ska ha identifierat och åtagit sig ett antal processförbättringsåtgärder som kommer att vidtas av teamet i nästa sprint.
- Frågor att besvara
 - Vad fungerade bra?
 - Vad fungerade mindre bra?
 - Vad borde vi sluta göra?
 - Vad borde vi börja göra?
- Tid för detta: efter sprint Review.



Bättre bild på sprint retrospective



Därefter börjar man om från början...

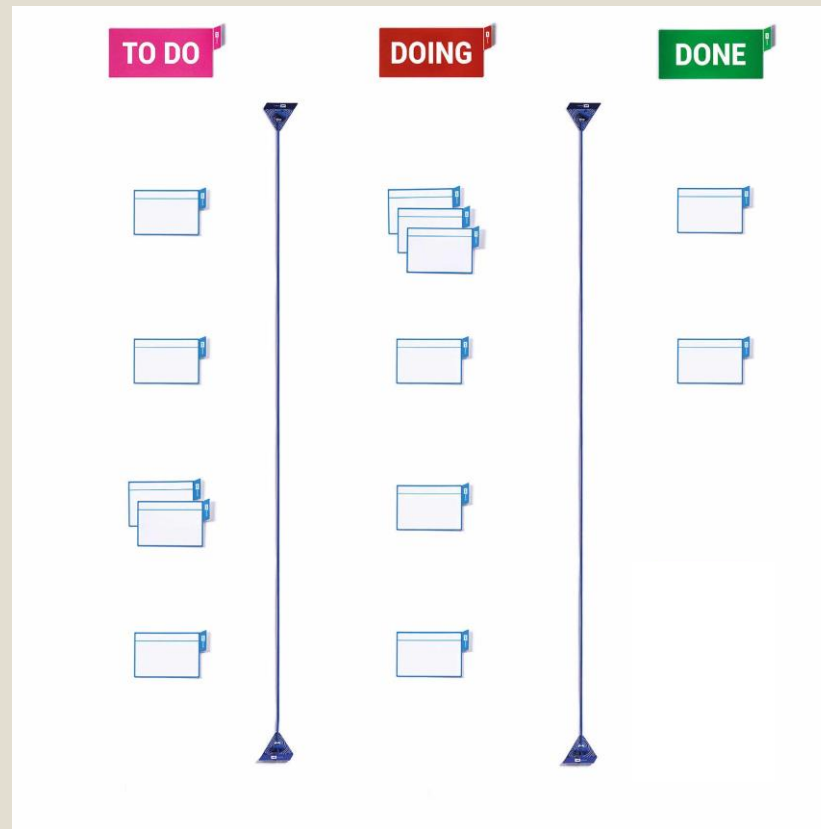


Kanban

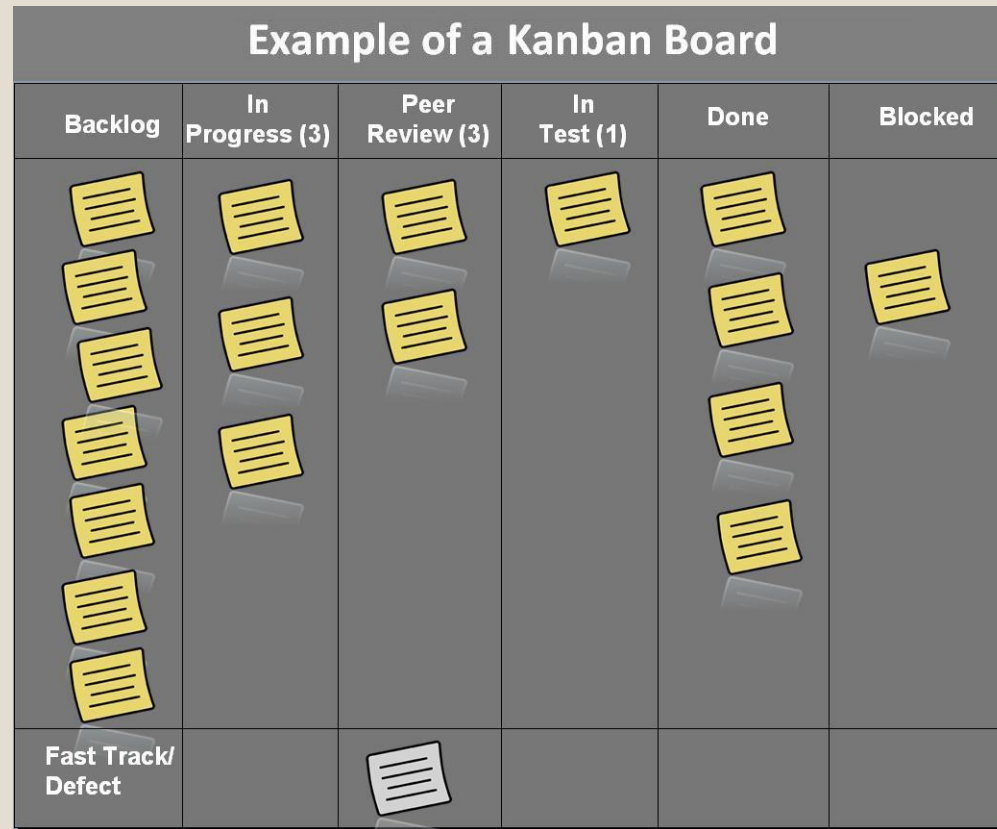


[This Photo](#) by Unknown Author is licensed under [CC BY-ND](#)

Kanban board



Kanban board



Kanban board

Pool of Ideas	Feature Preparation		Feature Selected	User Story Identified	User Story Preparation		User Story Development		Feature Acceptance		Deployment	Delivered
	3 - 10		2 - 5	30	15		15		8		5	
Epic 431	In Progress	Ready			In Progress	Ready	In Progress	Ready (Done)	In Progress	Ready		Epic 294
Epic 478	Epic 444	Epic 662	Epic 602			Story 802-02	Story 802-06	Story 802-08	Epic 401	Epic 609	Epic 694	Epic 386
Epic 562	Epic 589					Story 802-03	Story 802-04	Story 802-05	Epic 468	Epic 577	Epic 276	Epic 419
Epic 439	Epic 651		Epic 302	Story 302-05	Story 302-01	Story 302-07	Story 302-08	Story 302-09	Epic 362		Epic 339	Epic 388
Epic 329			Epic 335	Story 335-06	Story 335-09	Story 335-04	Story 335-05	Story 335-08			Epic 521	Epic 287
Epic 287				Story 335-08	Story 335-03	Story 335-08	Story 335-07				Epic 582	Epic 274
Epic 606			Epic 512	Story 512-04	Story 512-07	Story 512-02	Story 512-04					
				Story 512-05	Story 512-08	Story 512-08						
	Discarded											
	Epic 511	Epic 213										
	Epic 221											

Policy

Business case showing value, cost of delay, size estimate and design outline.

Policy

Selection at Replenishment meeting chaired by Product Director.

Policy

Small, well-understood, testable, agreed with PD & Team

Policy

As per "Definition of Done" (see...)

Policy

Risk assessed per Continuous Deployment policy (see...)

User story



Epics och initiatives



- Användarberättelser är byggstenarna i större agila ramverk som epics och initiatives.
- Epics är stora arbetsobjekt uppdelade i en uppsättning berättelser, och flera epics består av ett initiativ.
- Dessa större strukturer säkerställer att utvecklingsteamets dagliga arbete bidrar till de organisatoriska mål som är inbyggda i epics och initiatives.
- Definition av epic och initiatives:
[Epics, Stories, Themes, and Initiatives | Atlassian](#)

User story

- En användarberättelse är den minsta enheten för arbete i ett agilt ramverk.
- En användarberättelse är en informell, allmän förklaring av en programvarufunktion, skriven **ur slutanvändarens perspektiv**. Dess syfte är att artikulera hur en programvarufunktion kommer att ge **värde till kunden**.
- Det är frestande att tro att användarberättelser är, enkelt uttryckt, systemkrav. Men det är de inte.
- En viktig komponent i agil mjukvaruutveckling är att **sätta människor först**, och en användarberättelse sätter slutanvändare i centrum för konversationen. Dessa berättelser använder **icke-tekniskt språk** för att ge sammanhang för utvecklingsteamet och deras insatser. Efter att ha läst en användarberättelse vet teamet **varför** de bygger, **vad** de bygger och **vilket värde** det skapar.
- Användarberättelser är en av kärnkomponenterna i ett agilt program. De hjälper till att tillhandahålla ett **användarfokuserat** ramverk för det dagliga arbetet – vilket driver på samarbete, kreativiteten och en bättre produkt överlag.

User story

- Användarberättelser passar bra in i agila ramar som scrum och kanban.
- Inom scrum, användarberättelser läggs till i sprintar och avverkas under hela sprinten.
- Kanban-teamen drar in användarberättelser i sin backlog och kör dem genom sitt arbetsflöde.
- Det är detta arbete med användarberättelser som hjälper scrumteamet att bli bättre på tidsuppskattning och sprintplanering, vilket leder till mer exakta prognoser och större smidighet.

User story

- Användarberättelser tjänar ett antal viktiga fördelar:
- **Berättelser hålla fokus på användaren.** Att-göra-lista håller teamet fokuserat på uppgifter som behöver bockas av, men en samling berättelser håller teamet fokuserat på att lösa problem för riktiga användare.
- **Berättelser möjliggör samarbete.** Med slutmålet definierat kan teamet samarbeta för att bestämma hur användaren bäst ska betjänas och uppnå det målet.
- **Berättelser driver kreativa lösningar.** Berättelser uppmuntrar teamet att tänka kritiskt och kreativt om hur man bäst löser problemet för att nå ett slutmål.
- **Berättelser skapar drivkraft.** Med varje hanterad historia åtnjuter utvecklingsteamet en liten utmaningar och en liten vinst, vilket ger en drivkraft.

Hur man skriver användarberättelser

- Tänk på följande när du skriver användarberättelser:
- Definition av "Klar"
 - Berättelsen är i allmänhet "klar" när **användaren** kan slutföra den angivna uppgiften. Se till att definiera vad det är.
- Disponera underaktiviteter eller uppgifter
 - Bestäm vilka specifika steg som måste slutföras och vem som är ansvarig för var och en av dem.
- Användare - personas
 - För vem? Om det finns flera slutanvändare kan du överväga att göra flera händelser.
- Flera steg?
 - Skriv en berättelse för varje steg i en större process.
- Lyssna på feedback
 - Prata med användarna och fånga upp problemet eller behovet direkt från dem. Gissa inte berättelser om du kan hämta dem från dina kunder.
- Tid
 - Tidsåtgång är ett känsligt ämne. Många utvecklingsteam undviker diskussioner om tid helt och hållet, utan förlitar sig istället på sina uppskattningar.
 - En userstory bör vara möjliga att slutföra i en sprint. Berättelser som kan ta veckor eller månader att slutföra bör delas upp i mindre berättelser.
- När användarberättelserna är tydligt definierade, se till att de är synliga för hela teamet.

User story - mall och exempel

- Användarberättelser uttrycks ofta i en enkel mening, strukturerad enligt följande:
- **"Som [persona], [vill jag], [så att]."**
- Eller...
- As {persona}, I want {what?} so that {why?}.
-
- **[persona]:** Vem bygger vi detta för? Vi är inte bara ute efter en jobbtitel, vi är ute efter personens persona. Exempel: Max. Vårt team borde ha en gemensam förståelse för vem Max är. Vi har förhoppningsvis intervjuat massor av Max' s. Vi förstår hur den personen fungerar, hur de tänker och vad de känner. Vi har empati för Max.
- **[Vill jag]:** Här beskriver vi deras avsikt – inte de funktioner som de använder. Vad är det de faktiskt försöker uppnå? Den här satsen ska vara fri från tekniska begrepp – om du beskriver någon del av användargränssnittet och inte vad användarmålet är, har du missat poängen.
- **[Så att]:** hur deras omedelbara önskan att göra något som passar in i deras behov? Vad är fördelen de försöker uppnå? Vad är det stora problemet som behöver lösas?

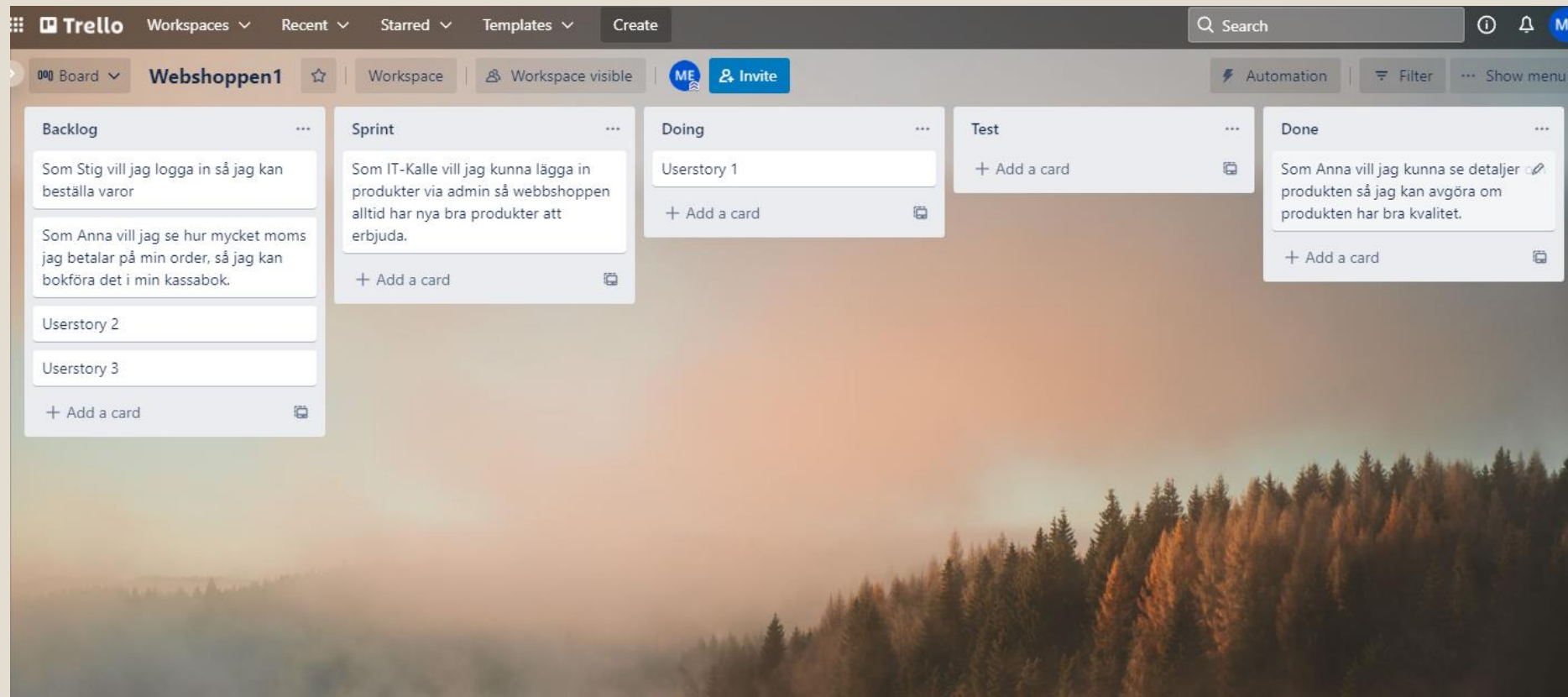
Persona Den ideala kunden

- En persona är en fiktiv representation av din ideala målgrupp
- Utgå från målgruppsparametrarna och lägger till följande frågor:
 - Personens utmaningar
 - Personens tankar
 - Personens agerande
 - Personens orosmoment
- Exempel: Kvinna i 35-årsåldern som bor i Stockholm, har en akademisk bakgrund med en inkomst över genomsnittet. Hon jobbar inom IT, är småbarnsförälder samt använder Facebook 3 timmar per dag. Ser ofta på TV och tränar sällan.
- Vad heter hon?
- Vilka utmaningar, tankar agerande och orosmoment har hon?
- Vad fyller hennes behov, med den app vi bygger?

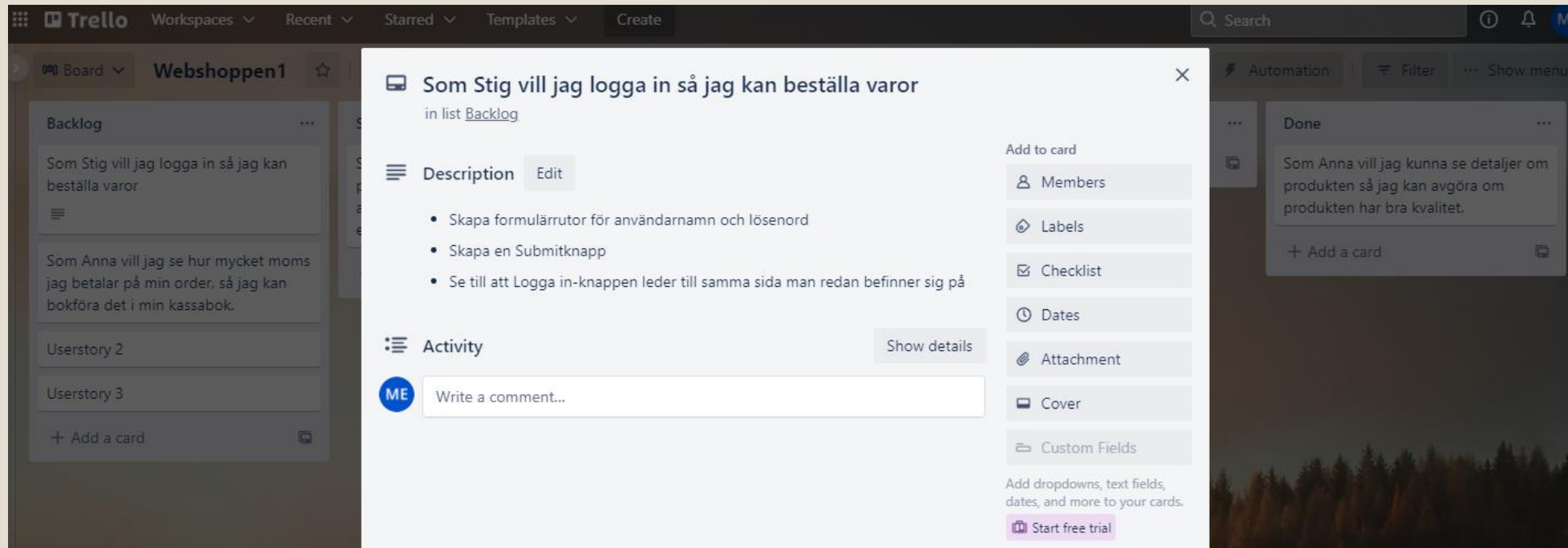
Exempel på user stories

- Som Max vill jag kunna bjuda in mina vänner, så vi kan njuta av den här tjänsten tillsammans.
- Som Anna vill jag organisera mitt arbete, så att jag kan känna mer kontroll.
- Som chefen Bertil vill jag kunna förstå mina kollegors framsteg, så jag kan bättre rapportera våra framgångar och misslyckanden.
- Som hamnägaren Sofia vill jag berätta om vilka fina faciliteter jag kan erbjuda, så mina kunder väljer min hamn.
 - Som hamnägare vill jag berätta om hur många duschar jag kan erbjuda, så mina kunder väljer att ta en dusch i min hamn.

Trello



Tasks



Trello

- Trello är ett enkelt projektverktyg som använder Kanban-tavla som planeringsverktyg.
- Trello ägs av Atlassian som också äger Jira, det kanske vanligaste projektverktyget på marknaden.
 - Jira är större och mer komplext än Trello
 - Trello räcker för våra ändamål.
- Trello är gratis, för små Teams
- <https://trello.com/>
- Demo: Kort intro till Trello
 - Skapa konto kan jag inte visa, kom ihåg att INTE välja free trial.
- Bra tutorial: [HOW to use TRELLO | ULTIMATE TRELLO TUTORIAL 2021 \[Project Management Software for BEGINNERS\] - YouTube](#)
 - Lägg gärna en timme idag på att alla i Teamet ser videon

Att tänka på

- Agila metoder:
 - Ska hjälpa till
 - Skapade FÖR utvecklare AV utvecklare
 - Ger överblick för alla
 - Löser inte problem, utan synliggör dem
 - Förutsätter att alla är med
 - Fasta punkter
 - Sprintstart
 - Dagliga standup-möten, max 15 min.
 - Sprint review
 - Sprint retrospect
 - Övrig tid: programmering i lugn och ro.

Uppgiften - Webbshoppa



Mockup

- Vad är en mockup?
 - I webbdesignens värld är en webbplatsmockup en (high-fidelity) simulering av hur en webbplats kommer att se ut. Webbplatsmodeller kombinerar strukturen och logiken i en skiss men med bilder, grafik och UI-element som slutprodukten kommer att ha.
 - I vår uppgift bygger ni en webbsajt som ser ut och upplevs fungera som webbshoppen ska göra
 - Men, ingen logik, interaktivitet, enbart bilder

Webbshoppen

- Ni ska “bygga” en webbshop som säljer ett valfritt sortiment.
- Webbshoppen ska ha följande funktionalitet
 - Startside
 - Välkomst-text och bild
 - Inloggningsruta med namn- och lösenordsfält
 - Tre utvalda produkter ska visas på sidan.
 - Shopsida med minst tre kategorier
 - T ex Tröjor, Byxor och underkläder (om ni gör en klädes-shop), cirka 5 produkter av varje.
 - Möjlighet att fritextsöka, formulärfält med Submit-knapp
 - Varje produkt ska vara klickbar för att få veta mer om produkten
 - Varje produkt ska ha en Köp-knapp som adderar produkten till kundkorg
 - Övrig info om varje produkt: Kort text om produkten, Pris , bild

Webbshoppen

- Varukorg
 - Valda produkter visas i lista
 - Knappar för att ändra antal
 - Möjlighet att ta bort product
 - Priset visas och summan av produkterna visas längst ner
 - Frakt-knapp
 - Namn, adress m m ska gå att fylla i i ett formular.
 - Val av frakt med minst två alternativ, med olika pris
 - Betala-knapp
 - Produkterna visas, med pris
 - Pris med frakt, samt moms visas.
 - Val av betalningsmetod, minst två. Betalningen behöver inte vara på riktigt

Webbshoppen - Admin

- Admin
 - Produkter – kunna lägga till nya och ändra
 - Namn
 - Text
 - Bild
 - Pris
 - Utvald
 - Lagersaldo
 - En sida med ett antal formulärfält, som går att fylla i, men inte sparas någonstans

Webbshoppen - Försäljningssiffror

- Butiksägarens sida
- En vy som visar Försäljning
 - Totalt
 - Per dag
 - Per kategori
- Lagersaldo
 - Lista på produkter som är slut
 - Produkter som det finns för mycket av

Webbshoppen - Teknik

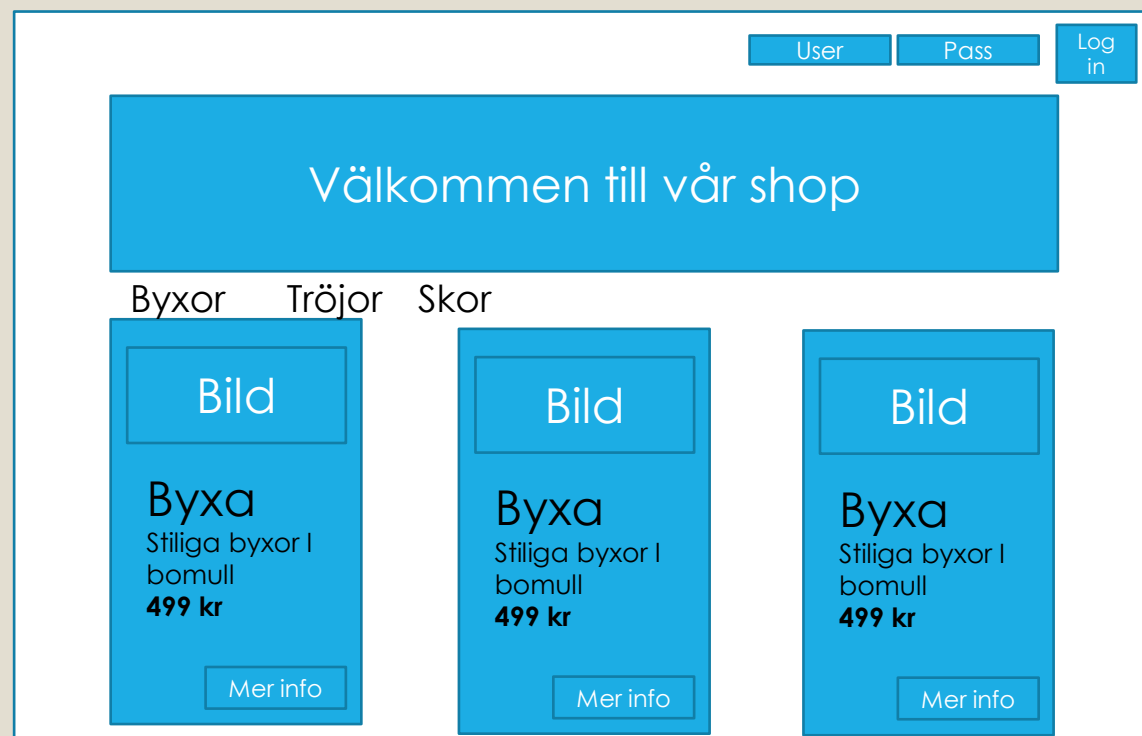
- Webbshoppen ska:
 - Vara “byggd” med enkelt ritprogram, och enklast möjliga “design”, främst för att placera sakerna på webbsidan
 - Se ut som en webbshop på ytan, men sakna interaktiv funktionalitet
- Design
 - Inga krav på snygg design.
 - Titta gärna på andra butiker på nätet för att få en känsla för hur de är upplagda och fungerar.
 - Använd gärna bilder på riktiga produkter

Mer info

- Projektet beskriver en klädbutik men det går bra att välja någon annan produkttyp.
- Projektets specifikation kan förändras (som i verkligheten)
- Förbättringsförslag mottages gärna
- Fokus är på den agila processen, inte hur webbshoppen ser ut. Den är främst till för att ha något att applicera agilt arbetssätt

Skissförslag

- Startside



Tidplan denna vecka

- Idag
 - Samlas i Gruppen, och gå igenom uppgiften.
 - Har alla förstått?
 - Några oklarheter?
 - Kontakta läraren gruppvis vid frågor.
 - Därefter...
 - **Skapa tre persona**
 - Kund
 - Administratör
 - Butiksägaren
- Nästa gång
 - Börja dagen med en daily scrum/standup
 - **Skapa User Stories** som täcker samtliga funktioner i kravspecifikationen, som fylls in i **Trello**
 - Lägg in alla utan någon ordning först, och placera därefter om dem i den ordning ni tycker att ni vill prioritera dem.
 - Ha som mål att få ihop cirka 20-30 userstories.
 - I Trello kan ni dela upp en userstory i mindre delar, så som Tasks. Detta kan ni göra antingen som separata "lappar" eller som punkter i Description
 - Bjud även in läraren till er Trello-board
 - Bestäm vilka kolumner ni vill ha i er Trello-board/Kanban-tavla
 - **Gör enkla skisser** på hur de olika vyerna ska se ut, med userstories i bakhuvudet.
 - Startsidan
 - Produktsida med en produktkategori
 - Detaljsida med en produkt
 - Kundkorgen
 - Leverans och betalningssida
 - Administrationssida
 - Butiksägarens lagersaldo
 - + en Tacksida, när man beställt
 - Visa skisserna för läraren under eftermiddagen

12 regler för agil planering

1. Involvera hela teamet
2. Planera på flera nivåer (smått och stort)
3. Håll storleks- och tidsuppskattningar åtskilda (valfritt)
4. Tänk på osäkerhet för funktioner och datum
5. Planera om ofta
6. Följ och annonsera framsteg
7. Var medveten om vikten av lärande
8. Arbeta med funktioner av rätt storlek
9. Prioritera funktionerna
10. Basera dina uppskattningar och planer på fakta
11. Planera inte för 100% av tiden (buffert, idealisk arbetsdag)
12. Samordna planering för att undvika beroenden

Frågor?

- Hör av er vid minsta oklarhet!
 - Gruppvis, så alla får samma information
 - Det går också bra att boka en tid under dagen
- Jag finns tillgänglig hela dagen

Länkar

- [What is Scrum?](#)
- [Kanban – Wikipedia](#)
- [HOW to use TRELLO | ULTIMATE TRELLO TUTORIAL 2021 \[Project Management Software for BEGINNERS\] - YouTube](#)

Länkar

- [10 Tips for Writing Good User Stories \(romanpichler.com\)](https://romanpichler.com/10-tips-for-writing-good-user-stories/)
- [What is User Story? \(visual-paradigm.com\)](https://visual-paradigm.com/what-is-user-story/)
- [How To Write User Stories, Epics, & Personas - Dev Life - YouTube](https://www.youtube.com/watch?v=8mUgD8v4Y04)
- [Scrum – Wikipedia](https://en.wikipedia.org/wiki/Scrum)
- [What is an Agile Developer? How is the Role Different? \(managedagile.com\)](https://managedagile.com/what-is-an-agile-developer-how-is-the-role-different/)
- [Product Owner Roles & Responsibilities for Software Projects \(concisesoftware.com\)](https://concisesoftware.com/product-owner-roles-responsibilities-for-software-projects/)
- [What is a Scrum Master? | Agile Alliance](https://www.agilealliance.com/glossary/scrum-master/)