



Campus Nyköping

SYSTEMUTVECKLINGENS GRUNDER OCH AGIL UTVECKLING 05

Mer Scrum och Kanban + Tidsestimering

Förra gången

- Vattenfall
- Agila manifestet
- SCRUM
- User Story och Persona
- Projektarbete presenteras: "Butiken"

Idag

- Upprop
- Mer SCRUM och Kanban
- Estimering
- Par/Mob-programmering
- Definition of Done
- MVP
- Grupparbete

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.

Projektets roller

- **Utvecklare**
- Produktägare
- Scrum master



SCRUM - Beståndsdelar

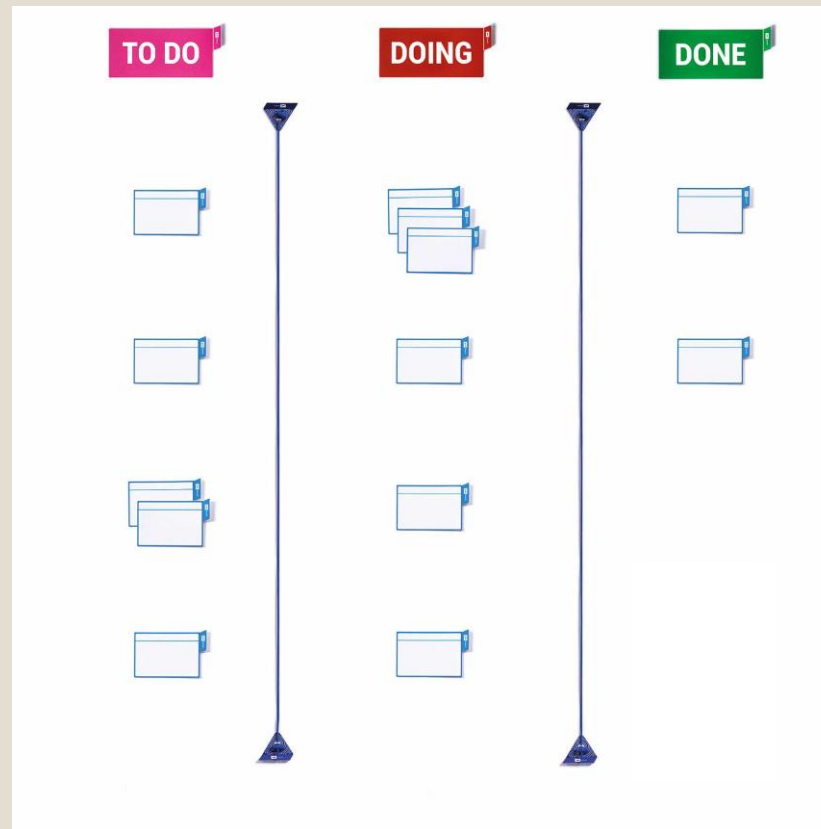
- Product backlog
 - En att-göra-lista. Här samlas alla önskemål som produktägaren vill ha.
- Sprintplanering
 - Planera vilka userstories vi ska lägga in i den här sprinten
 - För att kunna göra detta måste man estimerar/gissa hur lång tid varje userstory tar.
- Sprint backlog
 - En lista över det som förväntas hinnas med under **en** sprint
- Sprint
 - Arbetet delas in i sprintar. Varje sprint är mellan 3 och 30 dagar lång.
- Implementering
 - Själva utvecklingsarbetet
- 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.



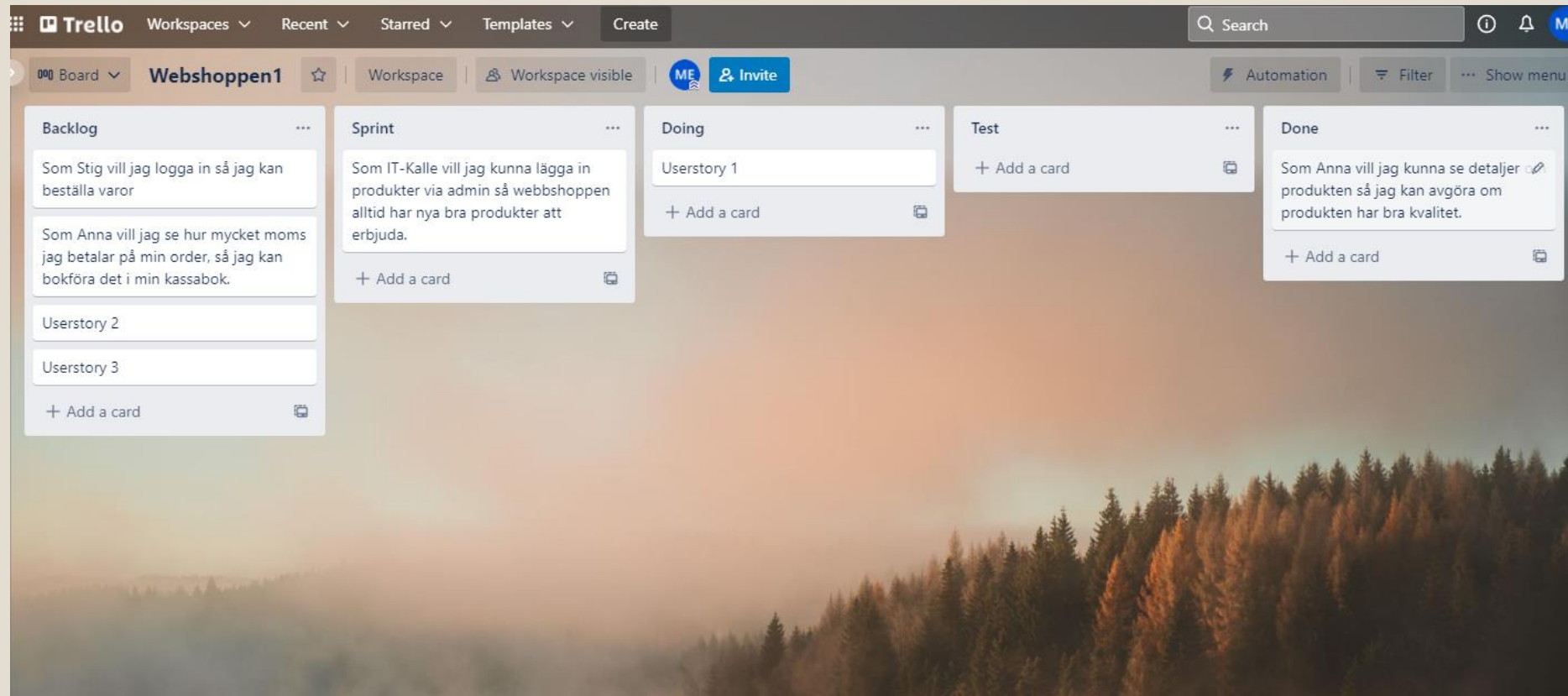
Produktbacklog

- 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

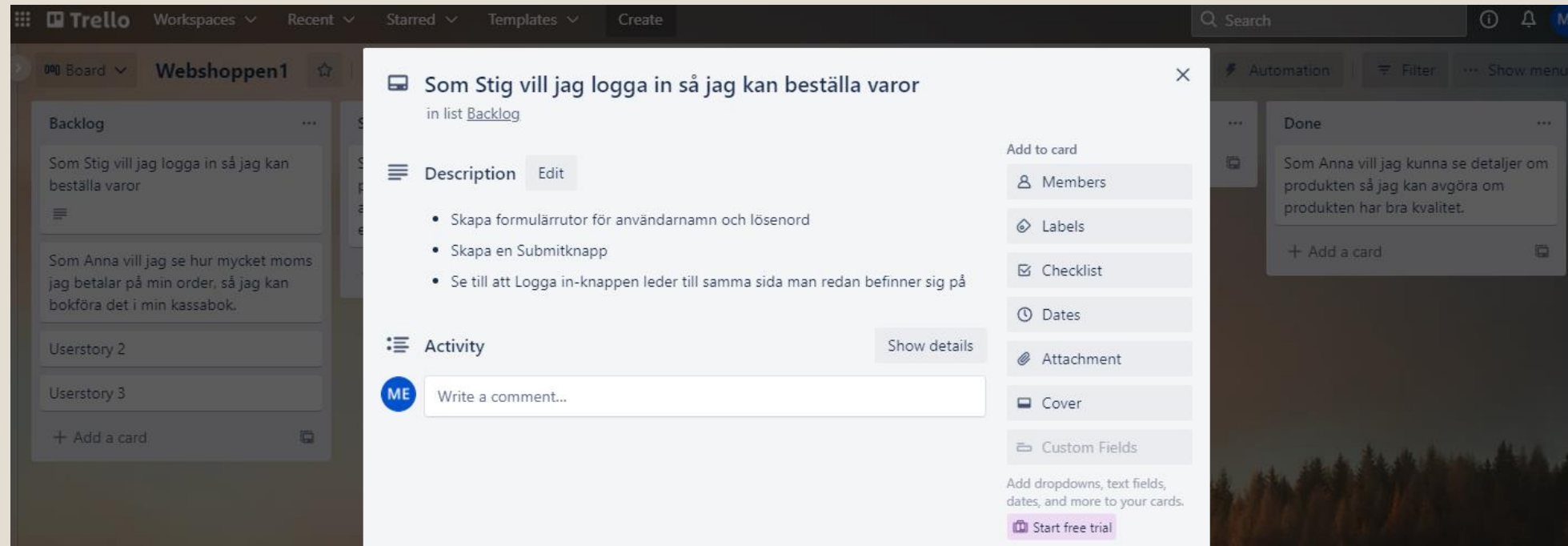
Kanban board



Trello



Tasks



Förslag på kanbankolumner

- Product Backlog eller bara Backlog
 - Här samlar man alla user stories i en enda lång rad.
 - De ska vara prioriterade efter hur viktiga de är.
- Sprint Backlog
 - Här lägger ni upp de user stories som ni tror att ni hinner med under sprinten
- Doing
 - Här placeras de stories som ni för närvarande jobbar med
- (Test)
 - När ni tycker att en story är klar, så läggs den i Test-kolumnen, så någon annan kan testa.
- Done
 - När userstoryn är klar så läggs den i Done-kolumnen.



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.



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.
- Utvecklingsteamet ger sedan en tidsuppskattning (vanligtvis i story points eller timmar) vad som krävs för att slutföra varje uppgift

Estimering – Uppskatta “tid”

- Hur lång tid tar det att läsa en 500 sidors bok?
- Hur lång tid tar det att köra till Umeå från Stockholm?
- Hur lång tid tar det att bygga ett sandslott?



- I takt med ökad komplexitet blir det nästa omöjligt att ge en exakt uppskattning.
- Uppskattning är en välgrundad gissning och det skiljer sig från det faktiska.
- Det finns ingen "exakt uppskattning".
- Varians är en verklighet så låt oss acceptera det!

Orsaker till osäkerhet vid uppskattning

- Görs av icke-utvecklare
 - Saknas kunskap om området eller tekniken
 - Finns ingen historik att lära sig ifrån
 - Oklara krav
 - Vad ska ingå?
-
- En del lean/agil-utvecklare säger t o m att tidsestimering är slöseri med tid.

Fördelar och tips

- Uppskattning hjälper till att förstå om produkten är livskraftig
- Uppskattning hjälper till vid projektplanering och genomförande.
- Uppskattning hjälper till vid prioritering.
- Det ger en annan möjlighet att samarbeta och få bättre förståelse för krav och design.
- Tips
 - Uppskatta i intervall i stället för att ge en enda siffra.
 - Uppskattning är INTE detsamma som löfte så ändra det när du lär dig något nytt.
 - Varje uppskattning är inte bättre än dess underliggande resonemang och antaganden.
 - Historiska data är bra för uppskattning.
 - Aktuella data är ännu bättre och det är där Agil utveckling fokuserar.

Ideal och verklig tid

- Ideal Tid är - Hur lång tid skulle ta om
 - Det är allt du arbetar med
 - Du har inga avbrott
 - Du har allt du behöver
- Förfluten/verklig tid är total tid det tar att avsluta arbetet
 - Tänk på hur många produktiva timmar om dagen du vanligtvis har.
 - Perfekt tid för en fotbollsmatch är 90 minuter – 2 x 45 minuter
 - Den förflutna tiden är mycket längre och svår att exakt förutsäga.
 - Ideal tidsuppskattning är lätt att uppskatta och förklara men svårt att omvandla till förfluten tid.
 - Utbildningar, möten, telefonsamtal och uppgiftsbyten etc påverkar skillnaden mellan idealtid och förfluten tid

Story points

- Story Point är en relativ storlek / storhet av uppgiften
 - Beror på "hur svårt det är"
 - Beror på "hur mycket det finns att göra"
- Relativa värden är det som är viktigt
 - En inloggningskärm är kanske 5; En sökfunktion är 8; en kundkorg är 35
 - Minsta storypoint kan vara 1, t ex en statisk sida med en text.
- Enhetslösa punkter
- Använd inte en enda guldstandard – Triangulera
 - Triangulering innebär att jämföra historien med flera andra berättelser.
- Varje team definierar dem som de vill. Vi kan inte jämföra uppskattningar av två olika team.
- Lättare att uppskatta, svårare att förklara
- Till en början kan t ex 6 points motsvara t ex en ideal arbetsdag, men målet är att det till slut inte ska motsvara en verklig tidsenhet, utan relativ.
- När ett team jobba tillsammans länge, börjar detta falla på plats.
- Då kan man också räkna ut hur många points man hinner med på en sprint.

Planning poker

- Ett sätt att estimerar hur svårt en story/funktion är
- De viktigaste stegen är:
 - Varje person i teamet har en hög med kort, varje kort har en siffra skriven på den (baserat på t ex icke-linjär skala)
 - Kund / produktägare läser en berättelse och det diskuteras kort
 - Varje person i teamet väljer ett kort som är hans eller hennes uppskattning
 - Korten vänds samtidigt så att alla kan se dem
 - Därefter diskuteras skillnaderna (särskilt de yttersta avvikelserna)
 - Därefter upprepas detta tills uppskattningar börjar bli likvärdiga
 - När man kommit till en gemensam siffra, så antecknas den
 - Produktägare eller kund men finns för att svara på eventuella frågor kring omfattning



Gemensam övning

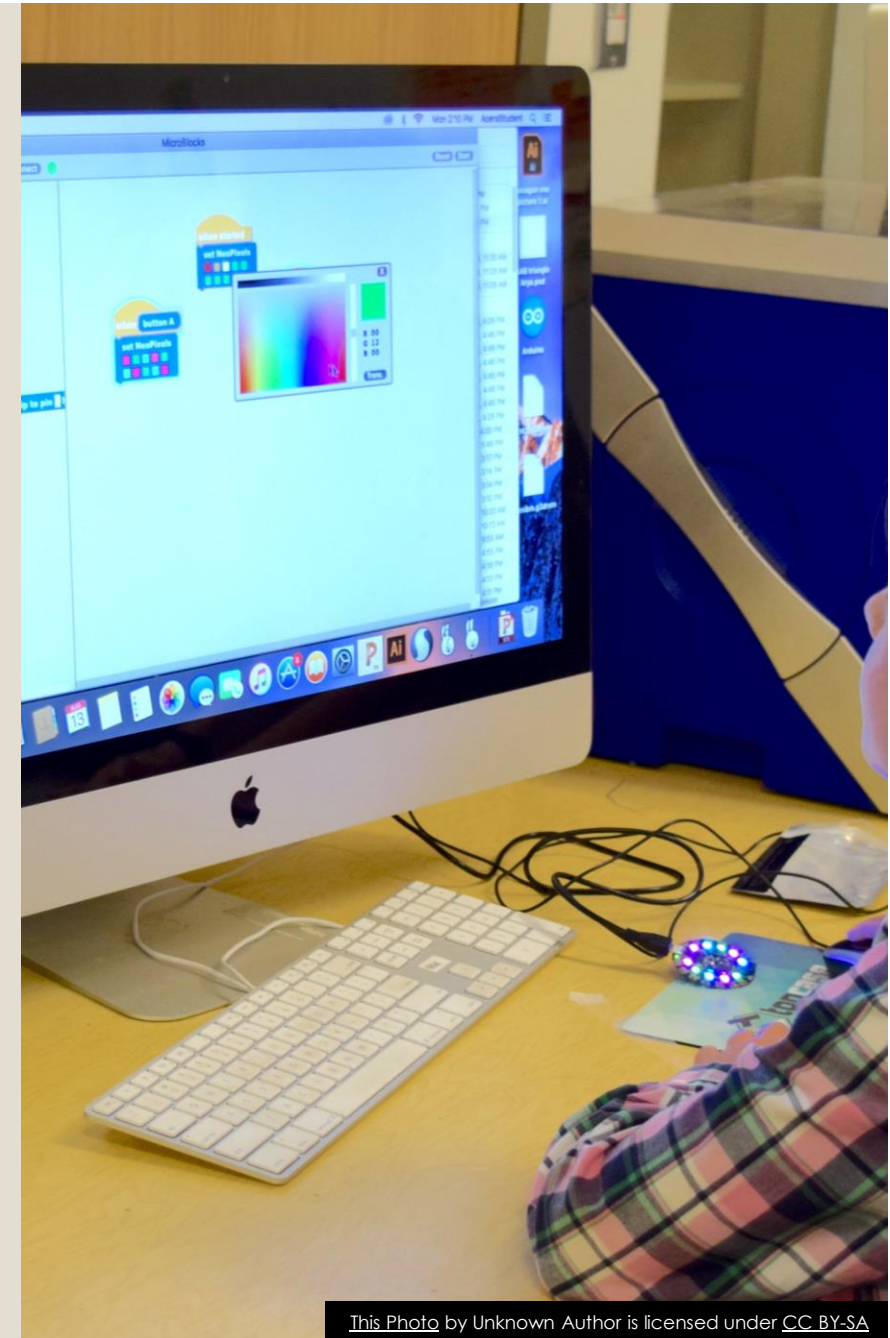
- <https://pokersizing.com/>
- I uppgiften tänker vi bara på UI-delen av arbetet
 - En poäng = en timme i vår övning, men normal saknar poängen enhet.
- Några olika funktioner
 - Placera en logotyp i applikationen, med länk till startsidan
 - En sökruta på en applikation om filmer
 - En meny med länkar till olika filmkategorier
 - Ett formulär för att registrera sig på en sajt
- Mer avancerat
 - En köpknapp, som då man klickar på den så sparas det man valt i en kundkorg
 - En knapp som tar dig till betalningsvyn, där det syns vilka betalningsmetoder vi vill använda.

Slutsats av övningen

- Den första iterationen kommer sannolikt att vara väldigt fel.
 - Oroa dig inte, lär dig och anpassa dig, korrigera dina uppskattningar
 - Ett team behöver tid för att börja träffa rätt.
 - Man pratar om teamets velocity
- “It’s better to be roughly right than precisely wrong.”
- Olika företag har olika upplägg på sina story-poäng, och även olika mellan olika team.

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.



Parprogrammering

- Parprogrammering är en agil arbetsmetod där två programmerare arbetar tillsammans på en arbetsstation. En, föraren, skriver kod medan den andra, observatören eller navigatören, granskar varje kodrad som den skrivs in. De två programmerarna byter roller ofta.
- Under granskningen beaktar observatören också den "strategiska" riktningen av arbetet, och kommer med förbättringar och påpekar sannolika framtida problem att ta itu med. Avsikten är att frigöra föraren att fokusera all sin uppmärksamhet på de "taktiska" aspekterna av att slutföra den aktuella uppgiften.
- Fallgropar:
 - *Frånvaro*: Om en i paret lutar sig tillbaka och exempelvis: tänker på annat, tittar på telefonen eller gör annat.
 - *Mäster*: Om den mer erfarna i paret "tar över" och kör utan att den andra hänger med så får man inte något bra samspel. Detta leder oftast till *frånvaro*.

Mob-programmering

- Mobbprogrammering, engelska: mob programming, kommer från tanken att hela "mobben"/svärmen/gruppen/teamet jobbar på ett och samma problem tillsammans, utan överlämningar och uppehåll i arbetet.
- I mobben finns det två roller, "driver" och "navigator". "Drivern" är den som styr datorn och rollen roteras mellan medlemmarna i fasta intervall. Tanken är att "drivern" enbart gör som den blir tillsagd av en "navigator".

"Navigator" är den som får styra "drivern" och samla input från de andra medlemmarna i mobben och bör vara någon som är insatt i problemet man försöker lösa.

- I vår mobb är alla utom "drivern" en "navigator", det vill säga att vem som helst kan ge instruktioner till "drivern". Det kräver lite mer självdisciplin än att bara ha en "navigator" men det får alla att känna sig inkluderade och har öppnat upp för många intressanta diskussioner som lett till bättre lösningar.
- Bra länk: [Vad är mobbprogrammering? | ProAgile](#)

Definition of Done

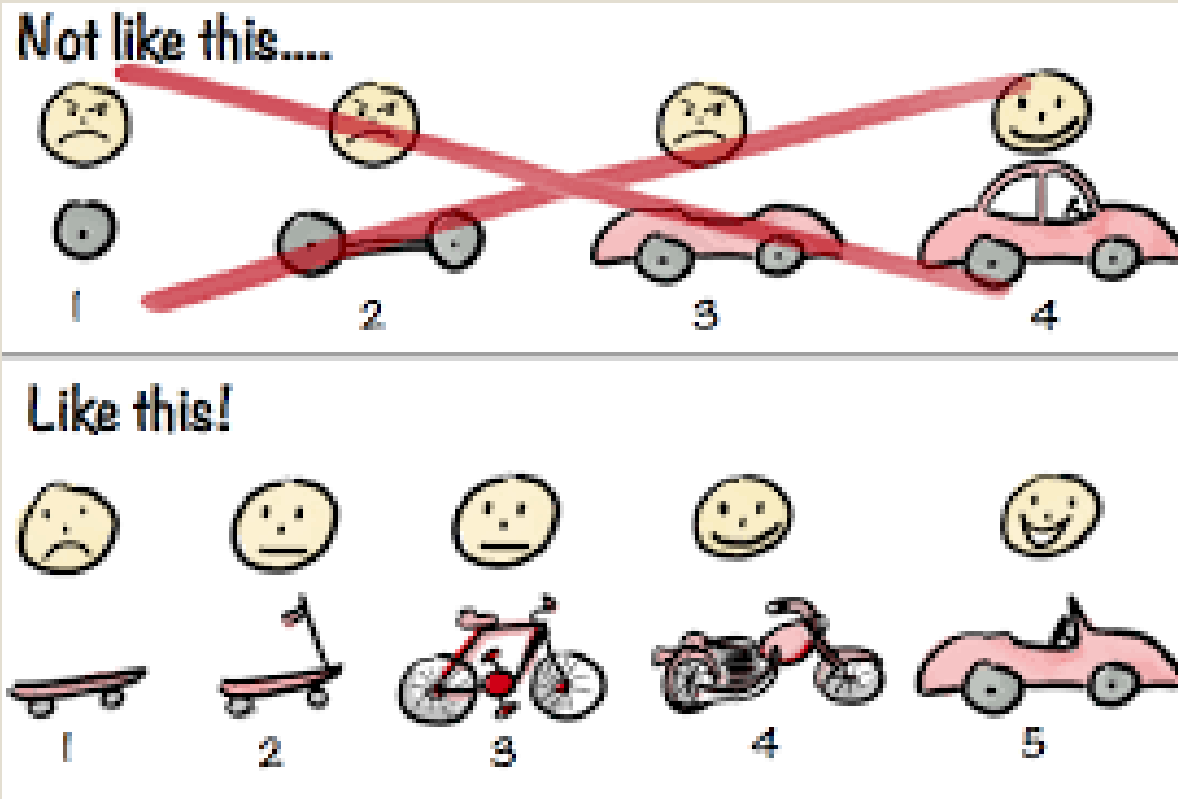
- När det som Scrum-teamet bestämde har blivit verklighet anses den vara "klart".
- Det slutförda arbetet ska vara av god kvalitet och kan levereras
- För en userstory så ska den tilltänkta användaren kunna göra det som bestämts, så den kan uppnå det han/hon vill.
- Userstoryn har blivit sann!



MVP – Minimum viable product

- Minimum viable product - minsta livskraftiga product
 - En minsta livskraftig produkt (MVP) är en version av en produkt med precis tillräckligt med funktioner för att kunna användas av tidiga kunder som sedan kan ge feedback för framtida produktutveckling.
 - har initialt tillräckligt mycket värde för att användare ska vilja använda eller köpa den.
 - uppvisar kommande fördelar för att fortsätta behålla tidiga användaren.
 - skapar en feedback-loop som guidar teamet vidare med utvecklingen.

MVP



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?





Sprint Review

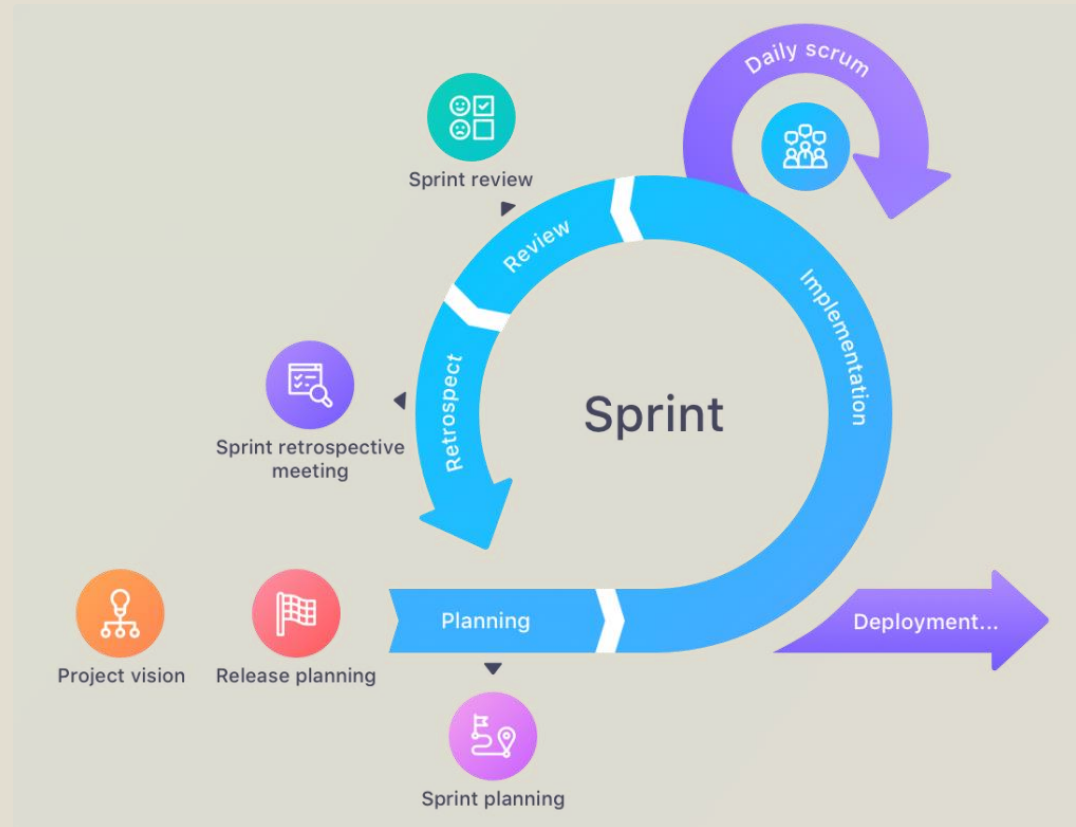
- Målet med denna aktivitet är att inspektera och anpassa den produkt som byggs.
- Reviewen fokuserar på att granska just de funktionerna som utförts under sprinten.
- Alla närvarande får tydlig insyn i vad som händer och har möjlighet att hjälpa till att påverka den kommande utvecklingen för att säkerställa 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 torsdag 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.



Därefter börjar man om från början...



Länkar

- [What is Scrum?](#)
- [Kanban – Wikipedia](#)
- [HOW to use TRELLO | ULTIMATE TRELLO TUTORIAL 2021 \[Project Management Software for BEGINNERS\] - YouTube](#)
- [10 Tips for Writing Good User Stories \(romanpichler.com\)](#)
- [What is User Story? \(visual-paradigm.com\)](#)
- [Kanban and Scrum - Making the Most of Both \(infoq.com\)](#)
- [What Is Kanban? Explained in 10 Minutes | Kanbanize](#)
- [https://pokersizing.com/](#)
- Plus dokumentet som ligger upplagt: Agile Estimation.pdf