

Systemutvecklingsprojekt med Scrum och eXtreme programming

Kursträff 3 – Planering och estimering

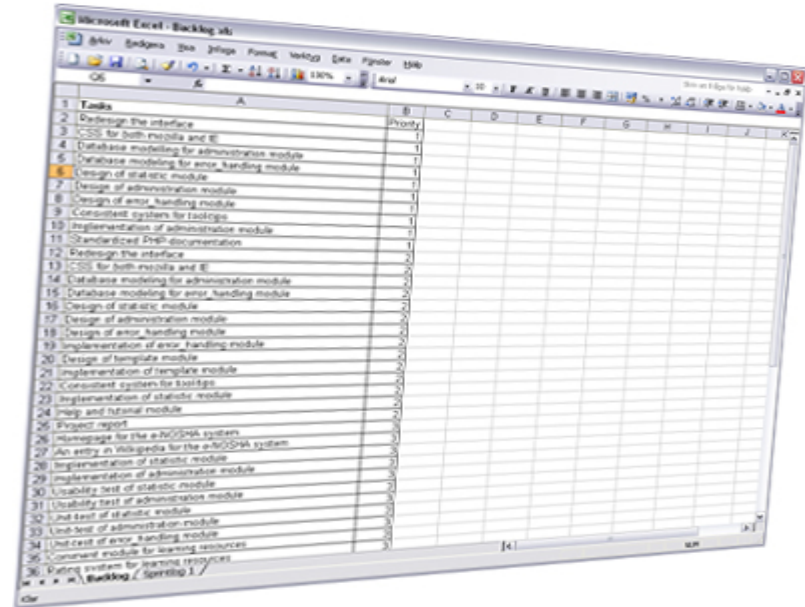
Lärare:

Mathias Hatakka - mathias.hatakka@oru.se

Andreas Ask - andreas.ask@oru.se

Artefakter

- Product backlog
- Release backlog
- Sprint backlog
- Burndownchart



Product backlog

- En kravlista som representerar allt som kan tänkas implementeras i systemet
- Allt som representerar jobb!
- Till en början ofullständig (baserad på en vision, en brainstorming session, dokumentation osv.)
- Dynamisk
- Prioriterad
- Innehåller *issues* som senare kan bli inkluderad i backloggen (ska även dessa prioriteras)
- Best-guesses och uppskattningar

Release backlog

- Den funktionalitet från Product Backlog som skall vara med i första releasen av applikationen.
- Ligger till grund för Sprint Backlog

Sprint backlog

- Funktionalitet som behövs för att uppfylla Sprint målet
- Detaljerad specifikation
- Varje uppgift skall ta 4-16 timmar att slutföra
- Sprint Backlog → är dynamisk och uppdateras genom hela Sprinten
(vad som är gjort, hur mycket tid saker tar, ny funktionalitet osv.)
- Endast Scrum Teamet får ändra i Sprint Backlog!

Release planning

- Skriv user stories (krav)
 - Tidsuppskatta user stories
 - Dela/kombinera user stories
- Fas 1
- Kund och team prioriterar
- Fas 2
- Förändringar på krav
 - Förändringar på uppskattningar
- Fas 3

User stories

User stories är små övergripande beskrivningar av krav med slutanvändarna i fokus.

Kommer arbeta med att tidsätta varje hög

Kommer arbeta med att prioritera varje hög

Klassa varje hög efter vikten för kunden (1-3)

Klassa efter risk:

- Fullständighet (1-3)

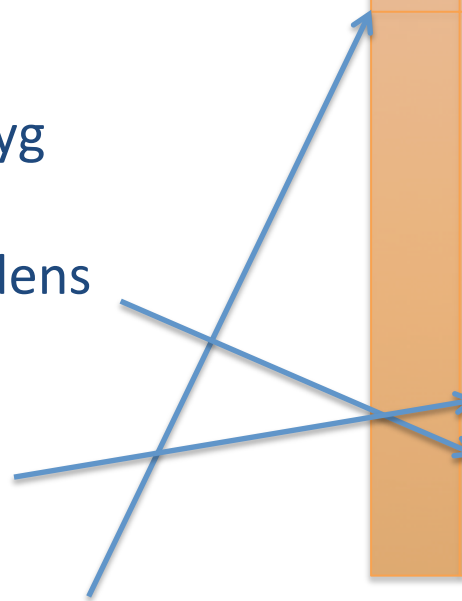
- För ändlighet (1-3)

- Komplexitet (1-3)

Att skriva user stories

- User stories skrivs av intressenterna
- Använd enkla verktyg
- Ange estimeringstidens omfattning
- Ange prioriteringen
- Ange unik identifierare

23. / Titel
<i>Studenter kan beställa examensbevis</i>
Prioritet: 3
Estimering: 5



Att skriva user stories

Exempel:

Som <roll> vill jag <mål/händelse>

- Som användare vill jag kunna söka efter information från varje sida

Som <roll> vill jag <mål/händelse> för att <syfte>

- Som lärare vill jag boka visning online för att besöka museet med min klass

Detaljera en user story

- User stories innehåller väldigt lite information, det finns tillfällen då dessa behöver mer förklaring innan man börjar arbeta med dem.
 - Under modelleringen med intressenterna
 - Under iterationsplaneringen
 - Under implementationen

Titel	
	<i>Som student vill jag beställa ett examensbevis för att kunna visa upp det till arbetsgivaren.</i>
	Prioritet: 3
	Estimering: 5

Bekräftelse/Förklaring:

Studenten måste erhålla godkända 180hp inom sitt ämnesområde

Studenten måste vara en nuvarande eller tidigare student

Studenten kan endast beställa två examensbevis i taget

Sand

- Att vara begravd i sand menas med att det finns flertalet user stories som inte är tillräckligt tydliga

Förutsättningar och planering

- Kunden ger:
 - a) funktionalitetsprioritet
 - b) scope
 - c) Datum
- Ofta finns ett slutdatum för driftsättning
 - Justera scope
 - Mindre viktig funktionalitet prioriteras ned
 - Var realistisk
- Skapa en release plan
 - När och vad

Estimering

Varför = För att ha möjlighet att göra förändringar när ny information tillkommer för en user story.

Story points

Varje **team** bestämmer vad en storypoint innebär för dem (i tid), förslagsvis: **en hel arbetsdag = 1 storypoint**. Därefter ses alla user stories över för att se ifall de har liknande estimeringar, "liknande uppgifter är lika stora".

Velocitet

- Idealet är 1 point = 1 vecka
- Ingen utvecklar 40 timmar i veckan!
- Underskattar ofta problemen
- Varje team har en velocitet (uppskattad tid/
verklig tid)
- Erfarna team har en bra uppskattning
- Justerad uppskattning: uppskattning/velocitet
- Skall mätas och uppdateras för varje iteration

- Velocity – totala antalet storypoints som blivit avklarade av teamet under en iteration. Ökar möjligheten att jämföra och planera nästkommande iteration.
- Om velociteten visar på mer arbete än planerat för den första iterationen fortsätter man med den uträknade arbetsinsatsen som visats efter den. Anledningen till missbedömningar är att uppskattningen från start inte riktigt stämt.

- Estimering vid par-programmering → storypoint beräkningen förblir densamma. Dock går det att beräkna storypoints utifrån "par-dagar".
- **Kom ihåg:** Gör det enkelt, lägg inte ner för mycket tid. Storypoints är en uppskattning av den beräknade arbetsinsatsen under en iteration. De kan variera från team till team och är inte huggna i sten.

Utvecklarnas ansvar: Definiera och uppskatta storypoints utifrån teamet, tillsammans med teamet.

Beställarens ansvar: Medverka under estimeringsmöten för att svara på utvecklarnas frågor vid oklarheter med de framtagna user stories.

Tre möjliga sätt att ta fram velociteten

1. Använd tidigare uppskattningar (att föredra om möjligt)
2. Påbörja första iterationen och använd den (inte alltid möjligt)
3. Gissa fram (behöver dock ha något att jämföra med)

Release planering

- Plan över när de olika leveranserna ska ske för projektet samt vad de ska innehålla, vilka delar i systemet – (delleveranser och slutleverans)
- Iterativ story-driven process är enkelt att ge förslag på leveranstillfällen, dock svårt att bestämma vad leveransen kommer att innehålla. Flera alternativa leveranstillfällen är det optimala (ex. **inom** oktober istället för **1**oktober).

Mäta velociteten

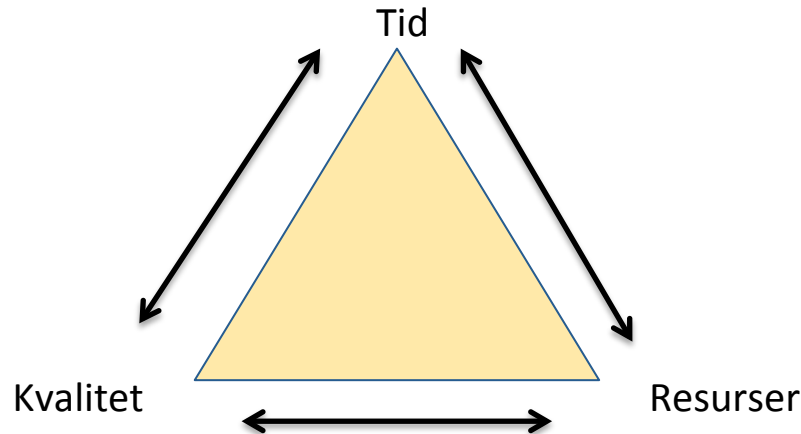
Release planeringens uppgift är att bryta ner projektet i **iterationer**. Varje iteration innehåller visst antal **story points**. Det totala antalet story points blir projektets **velocitet**.

- Teamets velocitet
- Iterationens velocitet
- Projektets velocitet

Prioritering

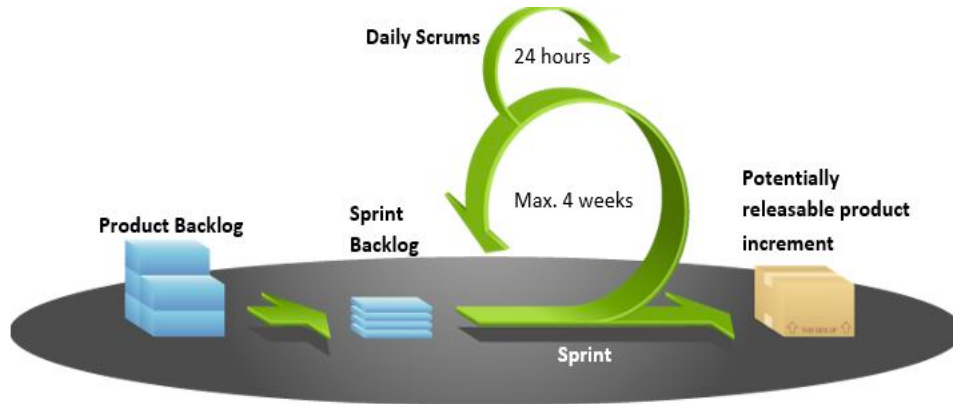
- User stories prioriteras utifrån risknivå för projektet, vanligtvis från hög-medel-låg.

"Måste ha" – Skulle behöva ha – Kan ha – Behöver inte den här gången"



Iteration

- Ingen förbestämd tid → bestäms av beställaren och utvecklarna. Vanligtvis **1-4 veckor**.
- Hellre något korta än för långa iterationer, så mycket som möjligt för att hålla. Kan komma tillfällen då iterationstiden behöver justeras.



Planera en iteration

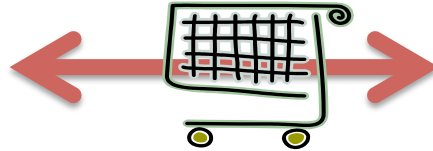
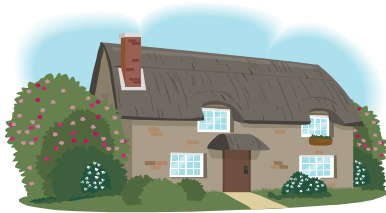
Med hjälp av release planeringen tar vi de samlade user stories och delar upp dessa till iterationer. Iterationsplanering görs med hela utvecklingsteamet (programmerare, testare m.fl.).

Varje user story ses över för att diskutera dess innebörd och innehåll. Därefter åtar sig teammedlemmarna sig uppgifter.

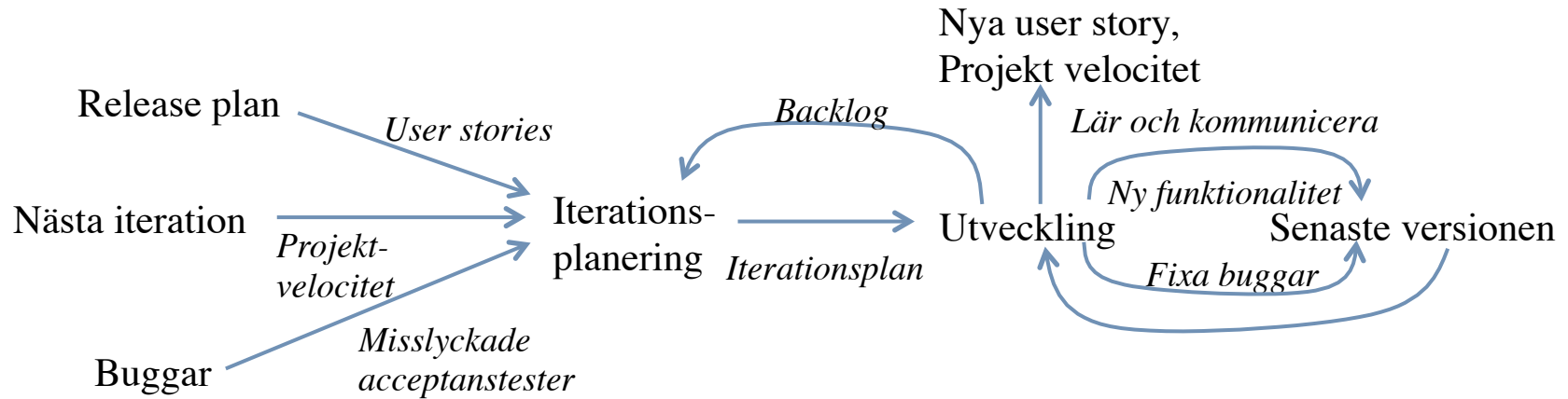
När varje user story diskuterats och blivit åtagen estimerar varje team medlem dessa individuellt.

Iterationer

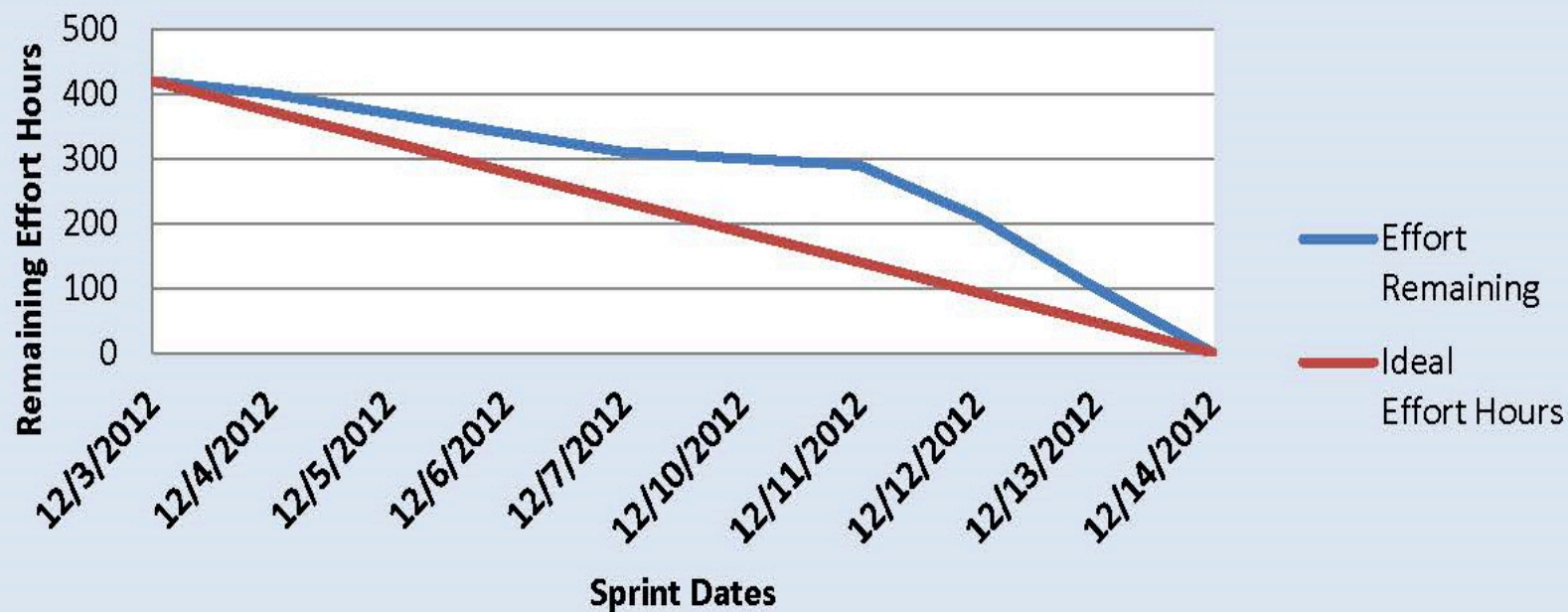
- Som flera små resor till affären
- Kan inte köpa allt man skall ha på en gång
- Behov = stories
- Kunden kan styra hur pengarna används



Iteration



Burn-Down Chart



Burndown chart

- Ett diagram över hur du och ditt team bränner igenom kundens user stories. Den visar det totala arbetsbördan i jämförelse till mängden av arbete som blir levererat för varje iteration.
- 1. Total arbetsmängd som planerats in i sprinten
- 2. Totala arbetstiden (i ert fall 1 vecka ex. tisdag-tisdag)

Definitioner

- User Story – en kort beskrivning i vardagligt språk av vad en användare vill uppnå
- Story points – den uppskattade utvecklingstiden i poäng (1 point = 1 vecka)
- Spikes – User stories som vi inte kan tidssätta
- Velocitet – uppskattad utvecklings tid

Sammanfattning

- Agila metoder "lättnvikts metoder"
 - Fokus → Scrum + eXtreme programming
- Scrum → Roller, Product backlog, Release backlog, Sprint backlog, Möten, Burndownchart, arbeta i **sprintar**.
 - Bygger på 5 Värderingar:
(Mod, Respekt, Engagemang, Fokus, Öppenhet)
- XP → Bygger på Värderingar, Principer och Praktiker
 - Använder User stories, Spikes, Velocitet, Planera iterationerna
 - (Planera på era veckomöten)
 - (ert fall – veckosprintarna)

