

TDP023 – Projekt: Agil systemutveckling

Johan Åberg
johan.aberg@liu.se

Tre moment

- Projekt – 8hp
- Marknadsföring av produkt – 2hp
 - Kopplat till projektarbetet
- Individuell rapport – 2hp
 - Kopplat till projektarbetet
- *Learning by doing and reflecting*

Tidsplan - översikt

- V4+5: Planering + förberedelse (sprint 0)
- V6+7: Sprint 1
- V8+9: Sprint 2
- V10+11: Sprint 3
- V13+14: Sprint 4
- V15+16: Sprint 5 + överlämning
- V18-21: Marknadsföring av produkt
- V18-21: Individuell rapport

Roller

- Lärare
 - Johan Åberg
- Studierektor
 - Jalal Maleki
- 5 utvecklingsteam
 - Ca 5 studenter per team
 - Extern kund (1 per team)
 - Intern scrum master för varje team
 - Externa coacher i VT1, två per sprint

Projekt med extern kund

- 5 projekt att välja mellan
 - Intresseanmälan görs på enkät efter denna föreläsning
- Projekt kan kräva underskrift av sekretessvillkor
 - Liknande villkor som för uppsatsarbeten med företagskunder och kursen Företagsprojekt
- Kunden äger det utvecklade systemet

Mer om rollerna

- ▶ **Kund.** Extern person som beställer arbetet. I denna kurs agerar kunden i princip som en *product owner* (i mån av tid).
 - ▶ Ser till att user stories finns i backlog (framförallt inför sprint start)
 - ▶ Prioriterar user stories inför sprint start
 - ▶ Deltar i demo vid sprint end och avgör vilka stories som får godkänt
- ▶ **Scrum master.** Är också utvecklare. Denna roll kan roteras inom gruppen.
 - ▶ Ansvarar för kontakten med kund
 - ▶ Ser till att burn-down-chart upprättas
 - ▶ Ansvarar för scrum-möten och att de blir till, samt uppdatering av burn-down-chart och tabellen med tasks och stories
 - ▶ Håller i sprint start
 - ▶ Håller i sprint end (demo + retrospective)
 - ▶ Fyller i sprintformuläret efter varje sprint
- ▶ **Utvecklare.**
 - ▶ Deltar aktivt i sprint start & sprint end
 - ▶ Deltar aktivt i scrum-möten
 - ▶ Utvecklar och testar valda tasks
 - ▶ Kommer i tid till planerade möten och utvecklingstider

Utvecklingsprocess

- Sprint 0
- Före sprint start
- Sprint start
- Dagligt scrum-möte
- Sprint end
 - Demo
 - Retrospective
- Scrum board

Sprint 0

Sprint 0

- ▶ Grupp: Utse en *scrum master*.
- ▶ Grupp: Gå igenom scheman och gör en detaljerad planering för vilka tider gruppen ska jobba. Bestäm ett lämpligt straff för sen ankomst (t.ex. bjuda gruppen på fika).
- ▶ Scrum master: Välj en lämplig yta på en stor whiteboard och rita upp en *scrum board* (se nedan).
- ▶ Grupp: Se till att ett möte med kunden äger rum. Utbyt kontaktuppgifter och kom överens med kunden om hur kommunikationen under projektet ska fungera.
- ▶ Kund: Delta i möte med projektgruppen och gör en översiktlig beskrivning av det planerade systemet (inklusive syfte, vision, etc). Eventuella sekretessavtal skrivs lämpligtvis under vid detta möte.
- ▶ Kund: Förbered user stories och förmedla till projektgruppen.
- ▶ Grupp: Genomför en tekniköversikt och riskanalys gällande val av teknikplattformar (utvecklingsverktyg, programspråk, databashantering, etc).

Före sprint start

Före sprint start

- ▶ Scrum master: Uppdatera backlog (justera och lägg tillbaka stories som fått fail).
- ▶ Scrum master: Se till att material finns på plats (pennor, post-its, etc).
- ▶ Kund: Förbered nya user stories (vid behov) och lägg dem i Backlog. Gör en prioritering av alla user stories i Backlog.
- ▶ Grupp: Tidsestimera alla user stories i Backlog i prioritetsordning.

Sprint start

Sprint start

- ▶ Grupp: Välj ut en mängd user stories (i prioriteringsordning) så att den uppskattade summan av tider stämmer ungefär överens med den tillgängliga tiden för user stories denna sprint (ta hänsyn till förra sprints velocity vid beräkning av tillgänglig tid).
- ▶ Grupp: Bryt ner valda user stories till tasks och skriv task-lappar.
- ▶ Kund: Det är bra om kunden finns tillhands och kan svara på frågor då stories bryts ner till tasks (antingen på plats eller via telefon/chat).
- ▶ Grupp: Tidsestimera alla tasks med **Planning poker**. Skriv den övernskomna tiden på respektive task-lapp. Om summan av tiden för alla tasks som ingår i en story är mycket större än den uppskattade tiden för storyn, kontakt kunden för en diskussion. Ska en annan story släppas från denna sprint? Ska storyns omfång (scope) definieras om? Ska prioriteringen av stories ändras? Allt är möjligt.
- ▶ Scrum master: Placera lapparna för alla valda stories på tavlan. Alla stories och deras tillhörande tasks placeras i kolumnen *Not Started* och sorteras i prioriteringsordning med högst prioritet överst.
- ▶ Grupp: Kör ett kort scrum-möte för att avgöra vem som börjar med vad. Varje gruppmedlem (eller par om man kör parprogrammering) väljer en task, och skriver sitt namn på lappen och flyttar den till kolumnen *Started*.
- ▶ Scrum master: Ritar upp ny burn-down chart: x-axeln speglar antal arbetsdagar som ingår i denna sprint. Y-axeln speglar summan av tidsuppskattningarna för alla tasks som hör till utvalda stories.
- ▶ Grupp: Utvecklingsarbetet börjar.

Planning poker

- Alla i teamet estimerar en story/task
 - Väljer ett kort/skriver en siffra
- Alla visar upp sitt val samtidigt
- Den som valt minst tid och den som valt mest tid diskuterar och enas om en estimering
- Finns varianter...

Dagligt scrum-möte

Dagligt scrum-möte

Mötet äger rum i början av varje planerad utvecklingsdag.

- ▶ Scrum master leder mötet och fördelar ordet.
- ▶ Grupp: Varje gruppmedlem besvarar kortfattat tre frågor: 1) Vad har jag gjort sedan förra mötet? 2) Vad åtar jag mig att göra till nästa planerad scrum möte? 3) Finns det något som hindrar mig i mitt åtagande (behöver jag hjälp med något)?
- ▶ Grupp: Alla gruppmedlemmar uppdaterar tidsuppskattningarna av de tasks de jobbar med. Om en task är färdig flyttas den till kolumnen *Done*. Om alla tasks för en user story är klara flyttas lappen med storyn till kolumnen *Ready for Review*.
- ▶ Scrum master: Uppdaterar burn-down chart så att den stämmer med de nya återstående tiderna.
- ▶ Scrum master: Om aktuell status i burn-down chart avviker väsentligt från ideallinjen, ska kunden kontaktas och informeras om läget. Tänkbara resultat från denna kontakt kan t.ex. vara att en eller flera user stories tas bort från denna sprint. Om så är fallet ska burn-down chart uppdateras så att den speglar det nya läget.

Sprint end - demo

Demo

Allokera en timme för demon. Schemalägg med kunden i god tid.

- ▶ Grupp: Demoa systemet för kunden, och gå igenom alla user stories som gjorts under sprinten.
- ▶ Kund: Avgör om det blir pass/fail för varje user story. User stories som fått fail skrivs om och läggs tillbaka i backlog.

Sprint end - Retrospective

Retrospective

Allokera en timme för retrospective. Äger rum efter demon (kunden deltar ej).

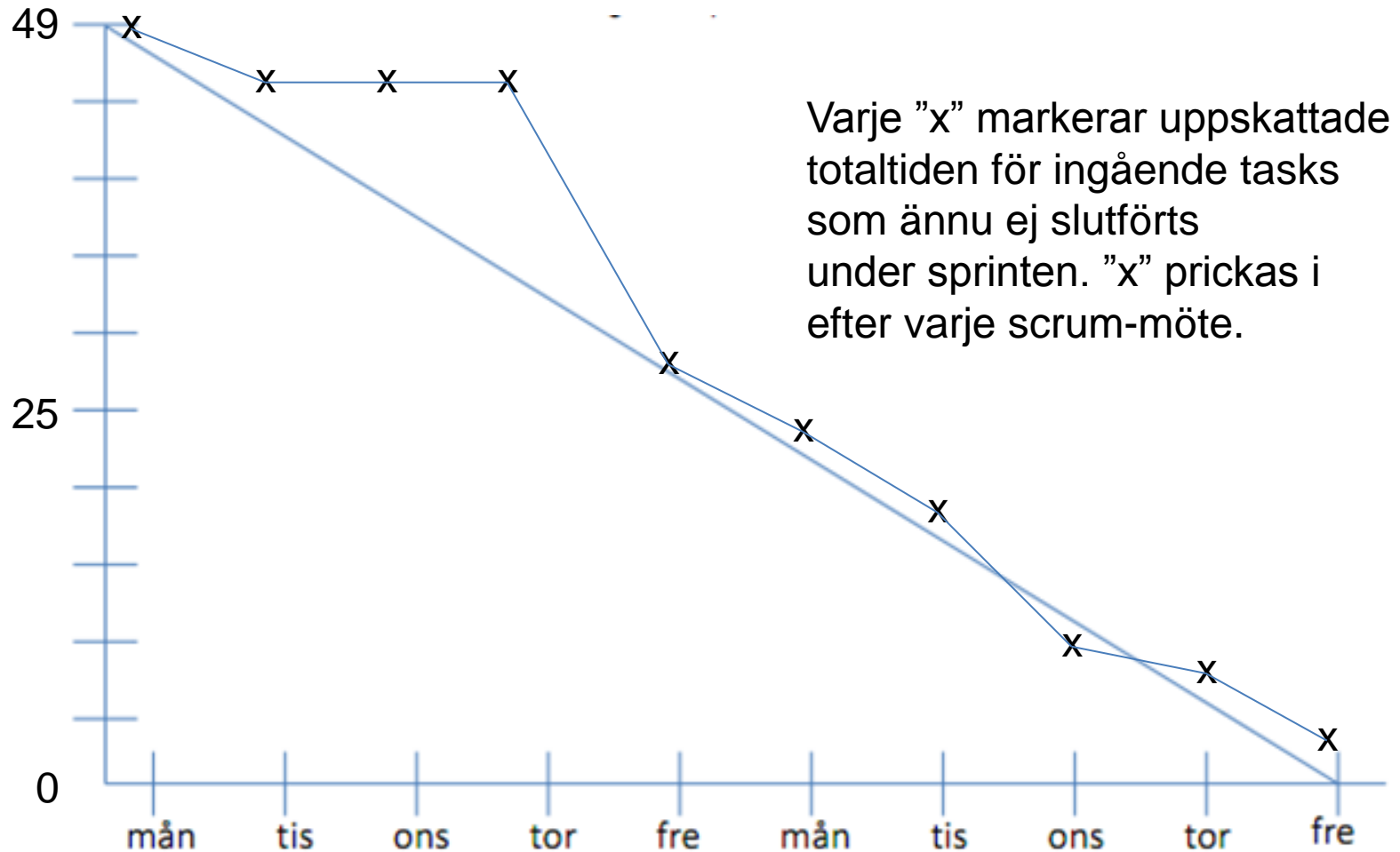
- ▶ Scrum master: Gå igenom sprinten som ägt rum och påminn om bra och dåliga saker som inträffat.
- ▶ Grupp: Varje gruppmedlem (inklusive scrum master) skriver (under tystnad) post-its med Plus (saker som gått riktigt bra under sprinten) och Delta (saker som inte fungerat under sprinten).
- ▶ Scrum master: Be varje medlem komma fram och sätta upp sina post-its på tavlan och kortfattat förklara vad som menas. Ingen diskussion.
- ▶ Scrum master: gå igenom alla positiva post-its på tavlan, summera och be om eventuella kompletterande kommentarer.
- ▶ Scrum master: gå igenom alla Delta post-its på tavlan, summera och diskutera igenom det hela med gruppen. Vad kan vi som grupp göra för att undvika dessa Delta?
- ▶ Grupp: Rösta om vilka Deltan som gruppen ska jobba med att förbättra under nästa sprint (tre röster per gruppmedlem). Tre Deltan väljs ut. Lapparna för dessa Deltan placeras på gruppens whiteboard. Dessa tas också upp för diskussion vid nästa Retrospective. Gick det bättre nu?
- ▶ Scrum master: Behåll lapparna för uppföljning.
- ▶ Scrum master: Efter retrospective rensar scrum master upp scrum board för att göra plats för nästa sprint, och beräknar den aktuella sprintens hastighet (velocity). Hastigheten beräknas som: $velocity = (\text{tillgänglig tid för sprint} / \text{summan av uppskattad tid för godkända user stories under sprinten})$. Denna hastighet används för att avgöra den maximala totaltiden för user stories som gruppen tar sig an vid nästa sprint. Scrum master fyller också i sprintformuläret och lämnar in till läraren.

Scrum board

Scrum board

- ▶ En scrum board ritas upp på en whiteboard och består av en burn-down chart och en progresstabell för user stories och tasks.
- ▶ Burn down chart uppdateras av scrum master efter varje scrum-möte. Diagrammet ska i varje tidpunkt spegla den totala uppskattade tiden för alla tasks som ännu ej flyttats till kolumnen *Done*. Burn-down chart uppdateras efter varje scrum möte.
- ▶ Progresstabellens kolumner är: *Not started*, *Started*, *Ready for Review*, *Done*. Det finns en rad per user story. Lappar med User stories står kvar i kolumnen *Not Started* som rubrik för tillhörande lappar med tasks, och flyttas sedan till *Ready for Review* och sedan till *Done*. Lappar med tasks förflyttas mellan *Not Started* till *Started* till *Done*. Rader och kolumner i tabellen kan t.ex markeras med tejp.

Burn down chart



Progresstabelle

Not Started	Started	Ready for Review	Done
			
			
			
			
			

Lappar för user stories & tasks

12 h

Som en användare ska jag kunna söka efter böcker med avseende på författarnamn.

2 h

Lisa & Klas

Lägg ut ett sökfält för författarnamn plus söknapp. Om markören är i sökfältet ska <return> ha samma funktion som att trycka på söknappen.

Tillgänglig tid

- Projektarbete (8hp)
 - Sprint 0: 4h schemalagd tid + 26h förberedelse & inläsning (totalt 30h)
 - Sprint 1-5: 20h schemalagd tid + 15h övrig tid (totalt 35h per sprint)
- Språklig kommunikation (2hp)
 - 52h per person
- Individuell reflektionsrapport (2hp)
 - 52h per person

Beräkning av tillgänglig tid för stories

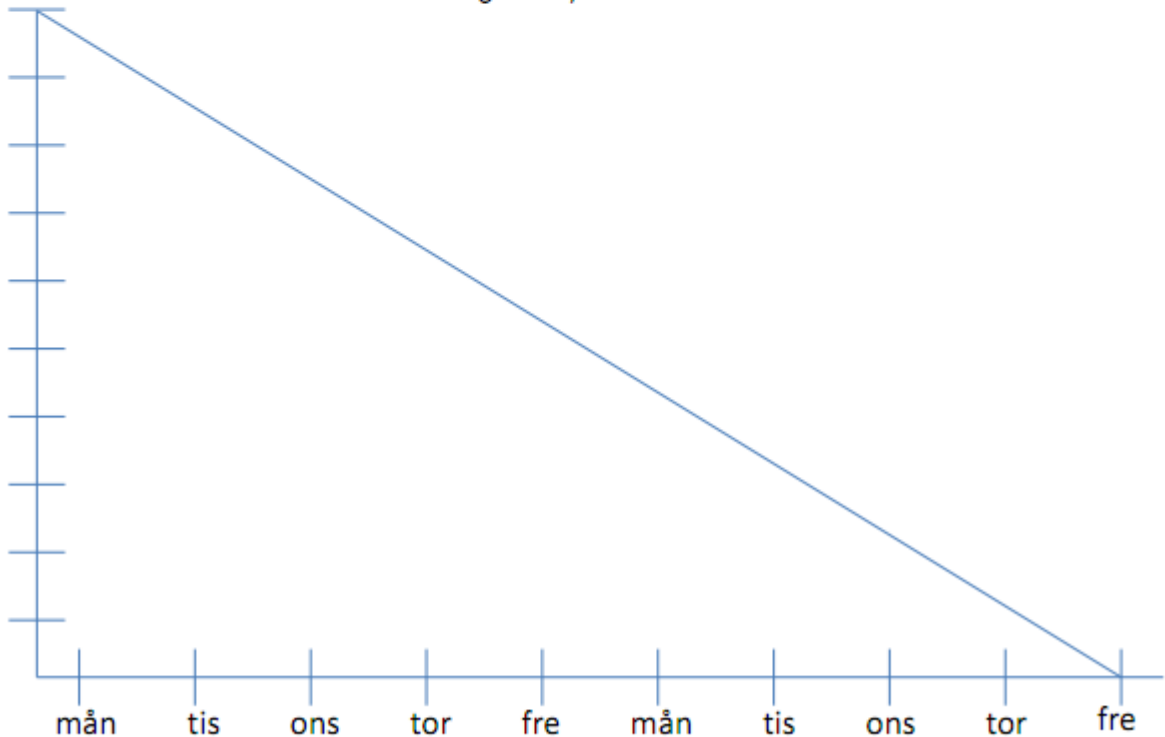
- Tillgänglig tid:
35h/pers/sprint
 - Antal pers: 4
 - Parprogrammering
 - 2 par
 - Velocity: 70%
 - Total tillgänglig tid för stories?
 - $(35 * 2) * 0.7 = 49h$
- Tillgänglig tid:
35h/pers/sprint
 - Antal pers: 5
 - Parprogrammering
 - 3 par
 - Velocity: 70%
 - Total tillgänglig tid för stories?
 - $(35 * 3) * 0.7 = 73.5h$

Beräkning av velocity

- Tillgänglig tid för stories, ej inräknat velocity
 - 70h
 - Uppskattad totaltid för *godkänd* stories
 - 38h
 - Velocity
 - $38/70 = 0,54$ (54%)
- Tillgänglig tid för stories, ej inräknat velocity
 - 105h
 - Uppskattad totaltid för *godkända* stories
 - 38h
 - Velocity
 - $38/105 = 0,36 =$ (36%)

Total estimerad tidsåtgång för stories denna sprint:	
Total estimerad tidsåtgång för godkända stories denna sprint:	
Tillgänglig tid denna sprint:	
Retrospective +	Delta

Burn-down chart – stora avvikelser från normallinjen ska förklaras, t.ex. på baksidan, tidsvärden ska anges för y-axeln



Daily scrum
Ja/nej

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--



Litteratur

- Kursbok

- Pieter Jongerius et al. Get Agile! Scrum for UX, design & development. BIS Publishers, 2013.

- Forskningsartiklar

- Williams, L., Kessler, R.R., Cunningham, W., Jeffries, R. Strengthening the case for pair programming, *IEEE Software*, 17(4), pp. 19-25, 2000.
- Jones, D.L., and Fleming, S.D. What use is a backseat driver? A qualitative investigation of pair programming. In *Proceedings of the IEEE Symposium on Visual Languages and Human-Centric Computing*, pp. 103-110, 15-19 September, 2013.
- Kai Stapel, Eric Knauss, Kurt Schneider, and Matthias Becker. Towards Understanding Communication Structure in Pair Programming. In *Agile Processes in Software Engineering and Extreme Programming, Lecture Notes in Business Information Processing Volume 48*, pp. 117-131, 2010.
- Kjetil Molokken-Ostvold, Nils Christian Haugen, Hans Christian Benestad. Using planning poker for combining expert estimates in software projects, *The Journal of Systems and Software* 81, pp. 2106-2117, 2008.
- Siva Dorairaj, James Noble, and Petra Malik. Understanding Team Dynamics in Distributed Agile Software Development. In *Agile Processes in Software Engineering and Extreme Programming Lecture Notes in Business Information Processing Volume 111*, pp. 47-61, 2012.

Examination

- Aktivt deltagande i projektarbetet (8hp)
- Språklig kommunikation (2hp)
 - Marknadsföringsmaterial + säljande presentation
- Individuell reflektionsrapport (2hp)
 - Deadline 26/5, 08.00
 - Del 1
 - Läs och sammanfatta/analyser tre forskningsartiklar
 - Del 2
 - Skriv en vetenskaplig diskussion som kopplar observationer från projektarbetet till de lästa artiklarna
 - OBS: För dagbok under projektets gång för att ha bra underlag för skrivandet

Etik

- Sekretess...

Frågor?

Uppdrag

- HTML5-app för iFiske
- Läraktiviteter i ett lärspele i historia för åk 4-6
- Utveckling av fotoapp och visualiserare till Brännskadeavdelningen BRIVA
- Editor för platsbaserad AR: från desktop till webbapplikation
- Stöd för att följa löpare i orientering: backend och app