

Programvaruteknik
Obligatorisk uppgift 1
Scrum och Lean Software Development

Hugo Källström, *oi11hkm*

April 2, 2014

Inledning

Vid utveckling av programvara krävs det inte bara expertis i form utav duktiga programmerare, utan hela processen från beställning till leverans spelar in för att få en slutgiltig produkt av god kvalitet. För att åstadkomma detta finns en hel del olika sätt att arbeta inom en organistaion eller ett företag, i den här artikeln kommer två sådana metoder att diskuteras.

Den första kallas *Scrum* och är en sorts arbetsprocess som beskriver hur man skall gå till väga för att utveckla programvara på ett effektivt och smidigt sätt. ”Scrum är ett ramverk för att utveckla och underhålla komplexa produkter” [1].

Den andra metoden som ska diskuteras kallas *Lean Software Development*. Detta är egentligen ingen arbetsmetod eller process utan snarare en definition på en utvecklingscykel. En mjukvaruprocess livscykel kan sägas vara ”lean” om denna följer grundpelarna från ”Lean Software Development”-rörelsen och principerna från ”Lean Software Development” [2].

Dessa två sätt att arbeta har tagit fram nya metoder för mjukvaruutveckling och är relativt nya. De mer traditionella ”steg-för-steg”-metoderna har blivit allt mer ovanligare då dessa inte lämpar sig särskilt bra för mjukvaruutveckling då de bland annat medför hög risk för fel och sämre kundkontakt under processen.

Agil och *Lean* programvaruutveckling råder bot på dessa problem genom att behandla små problem i taget och låta utvecklarna själva ta mer ansvar. Genom att bygga systemet i små inkrement där man hela tiden låter kunden komma i kontakt med utvecklingsprocessen för att ge tydligare direktiv och krav kan man minimera risken för fel och effektivisera hela utvecklingsprocessen. Det finns också större möjlighet för utvecklare att få och ge feedback under arbetet. Detta medför ”transparens” under utvecklingsprocessen, alla inblandade vet vad som ska göras och vad som gjorts. Det blir också lättare att ta beslut i och med denna ”transparens”.

Scrum

”A scrum (short for scrummage) is a method of restarting play in rugby football.”[3] Scrum är alltså en regel inom football som beskriver att spelet startas om när bollen är ur spel. Hur kan detta tillämpas inom mjukvaruutveckling?

Scrum kom till i början på 1990-talet och är ett sorts ramverk som beskriver hur en utvecklingsprocess ska gå till. Grundpelarna inom scrum lyder[1]:

1. Lättviktigt
2. Lätt att förstå
3. Svårt att bemästra

Det ramverk som beskriver scrum är alltså kort och koncist samt lätt att förstå sig på, men desto svårare att applicera i verkligheten. I stora drag består ramverket av ett *scrumteam* och dess roller, aktiviteter, artefakter, och regler.

Roller

Ett *scrumteam* består utav en produktägare, en scrummästare och ett utvecklingsteam. Produktägaren är den som ansvarar för att arbetet görs så effektivt som möjligt. Det är bara produktägaren som får ändra i produktbackloggen, det vill säga besluta vilka mål som ska sättas samt när och hur arbete som ska utföras. För att scrum ska lyckas måste utvecklingsteamet respektera produktägaren, det är dennes beslut som gäller och det reflekteras i produktbackloggen.

Scrummästaren är en del av utvecklingsteamet. Det är scrummästaren som ser till att utvecklingsteamet följer det ramverk som scrum står för, men också informerar utomstående om det arbete som görs inom teamet. Scrummästaren för en aktiv dialog med produktägaren om vilka mål som behöver sättas upp, hur teamet kan arbeta med dessa mål och även hur man arbetar ”agilt”.

Själva utvecklingsteamet består av ett antal personer med olika kompetens inom ett flertal områden, beroende på typen av arbete som ska utföras. Ett optimalt utvecklingsteam består av mellan 3-9 personer. Ett mindre team blir oeffektivt då man inte kan få lika mycket feedback och ta lärdom av varandra, ett större team blir för komplext och det blir svårt att tillämpa scrum.

Produktbacklogg

Produktbackloggen beskriver alla de krav som finns på produkten, den uppdateras kontinuerligt så länge produkten existerar. Det är produktägaren som bestämmer vad som prioriteras i produktbackloggen, men produktägaren för även en dialog med scrummästaren om vad som kan ge mest värde i produktbackloggen.

Tack vare produktbackloggen kan utvecklingsprocessen ske med små inkrement där man fokuserar på ett eller några krav i taget, gör klart dessa och sedan möter upp med kunden för att ta fram nya krav. Detta medför liten risk då det

är lätt att göra om eller lägga till krav utan att behöva ändra hela systemet. Kunden får även se hur produkten formas från start till slut.

Sprints

Det är under så kallade ”sprints” som själva arbetet utförs i scrum. Utvecklingsteamet väljer det som är högst prioriterat i produktbackloggen och skapar en ny sprintbacklogg från detta. I sprintbackloggen finns alla de krav som behöver uppfyllas för att klara av sprintmålet. En sprint kan vara från 1-2 veckor upp till cirka 1 månad. Kortare tid än så brukar vara ineffektivt, och under en längre tidsperiod kan kunden ändra sig, eller nya, viktigare, krav kan dyka upp.

Under en sprint kan utvecklingsteamet fokusera på det arbete som göras och behöver inte ta hänsyn till andra krav än de som finns i sprintbackloggen för tillfället. Det är också utvecklingsteamet som ansvarar för sprintbackloggen, teamet kan själva lägga till och ta bort krav om de anser att det behövs för att uppfylla sprintens mål.

En viktig del i scrum är att alla inblandade har en gemensam bild av vad ”klart” betyder, detta för att kunna markera uppgifter i både produktbackloggen och sprintbackloggen som ”klara”.

Möten

Scrum beskriver ett antal olika möten som ska hållas innan, efter och under en sprint. Dessa möten är till för att föra en dialog med produktägaren samt ge och ta feedback från utvecklingsgruppen.

”Sprintplanering” sker innan en sprint påbörjas. Detta är ett möte som under en enmånadssprint håller på i cirka 8 timmar. Här är hela scrumteamet delaktiga och tar fram de krav och mål som behövs för att uppfylla sprintmålet.

Ett ”Dagligt scrummöte” är som namnet antyder ett dagligt möte där utvecklingsteamet och scrummästaren träffas för att gå igenom vad som ska göras det närmsta dygnet. Teamet tar även upp vad som gjorts föregående dag och vad som behöver göras för att uppnå sprintmålet.

”Sprintgranskning” sker i slutet av en sprint och då träffas hela scrumteamet för en sammanfattning av vad som gjorts under sprinten. Detta kan ses som en statusuppdatering där teamet går igenom vad som ska ändras på produktbackloggen efter avslutad sprint. Mötet avser också att komma fram till vad som ska göras under nästa sprint.

Lean Software Development

Lean Software Development är till skillnad från Scrum inget strikt ramverk för hur en utvecklingsprocess ska se ut, det är snarare en filosofi om hur en organisation ska gå till väga för att på bästa sätt effektivisera sin verksamhet. Många av de riktlinjer som finns i Lean Software Development har *agil*-utveckling anammat. Det finns ett antal riktlinjer som definierar om en utvecklingsprocess är ”lean” [6].

Eliminera skräp

Betyder att allt onödigt arbete som inte bidrar till slutprodukten är skräp (waste). För att minimera detta ges utvecklingsgrupper eget ansvar så att de själva kan bestämma vad som ger ”värde” till slutprodukten.

Bygg för kvalitet

För att minimera fel ska dessa adresseras direkt och inte läggas på hög för att fixas vid ett senare tillfälle. Test Driven Development (TDD) är en bra metod för att minimera fel under utvecklingsprocessen. Men man kan också tillämpa till exempel parprogrammering.

Skapa kunskap

Här är feedback en viktig del, utvecklarna måste reflektera över vad som gjorts men även planera arbeten i framtiden. Eget ansvar spelar också en stor roll här, ett team ska inte bli tillsagda vad de ska göra utan snarare själva hitta en lösning på ett problem.

Skjut upp åtaganden

Att från början ha en fullständig produktspecifikation kan både vara bra och dåligt. Lean Software Development förespråkar dock att utvecklingen sker i små inkrement samt att val och åtaganden skjuts upp så mycket som möjligt.

Leverera snabbt

Om utvecklingsteam själva får bestämma vad de klarar av att leverera inom en tidsram har detta visat sig gynna den totala tiden det tar för produkten att bli klar. Lean Software Development uppmuntrar till att utvecklingsteam organiseras och tar ansvar själva.

Respektera andra

Återigen är fokus på att respektera utvecklingsteam och låta de ta ansvar över sitt eget arbete.

Optimera helheten

För att vara framgångsrik behöver helheten tas i beaktning, detta betyder att organisationen inte bara måste kolla på den enskilda produkten som utvecklas utan hela marknaden.

Utvärdering

Nedan följer en utvärdering av både Scrum och Lean Software Development där ett antal kriterier jämförs.

Table 1: Jämförelse av Scrum och Lean Software Development (LSD)

Kriterie/Metod	Scrum	Lean Software Development
Industri	Mjukvaruutveckling	Mjukvaruutveckling
Omfattning	Uvecklingsprocessen	Hela organisationen
Arbetsätt	"Sprints"	Kontinuerligt flöde
Ansvar	Utvecklingsteam	Utvecklingsteam

Industri

Scrum (som är en *agil*-utvecklingsmetod) är anpassad endast för mjukvaruutveckling, denna metod togs fram i februari 2011[5] av ett team i Chigago. Syftet var att hitta en ny metod som skulle rikta sig mot mjukvaruutveckling och göra utvecklingsprocessen mer effektiv. Från denna metod har ett antal ramverk skapats, och ett av dessa är Scrum.

Till skillnad från Scrum och *agil*-utveckling är "Lean" en metod som från början togs fram av Taiichi Ohno och Shiego Shingo på 1950-talet för Toyota[5]. Men under tidigt 1990-tal skrev James Womack och Daniel Jones ett antal böcker som hjälpte många andra industrier att följa den filosofin som "Lean" står för. Redan 1993 introducerade Robert "Bob" Charette "Lean Software Development" som visade på att det även gick att implementera Lean inom mjukvaruutveckling.

Att "Lean" från början inte applicerades på just mjukvaruutveckling kan vara bra att ha i åtanke när dessa två metoder jämförs.

Omfattning

Scrum är ett sorts ramverk som beskriver hur utvecklingsprocessen ska gå till i en organisation. Detta måste följas om en organisation vill använda sig av Scrum fullt ut. Lean Software Development är tvärt emot Scrum inget ramverk, utan en utvecklingsprocess inom en organisation kan anses vara "Lean" om denna följer samma filosofi som Lean Software Development står för.

Detta betyder att en organisation kan sträva efter att arbeta med Lean Software Development inte bara i utvecklingsprocessen utan även andra delar inom organisationen.

Arbetsätt

Sprints är en stor del av Scrum och det är inom dessa sprints som den största delen av arbetet utförs i utvecklingsprocessen. Detta betyder att arbetet sker i små etapper, och det är väldigt ovanligt att en sprint avbryts (det kan till och med vara dåligt för scrumteamet)[1].

Lean Software Development försöker istället hitta ett ”flow” i utvecklingsprocessen där man inte fokuserar lika starkt på sprints, men detta betyder inte att arbetet inte sker med inkrement (något som visat sig vara optimalt när det kommer till mjukvaruutveckling[2]). Detta visar också på att Lean Software Development inte fokuserar lika mycket på utvecklingsprocessen utan på helheten i organisationen.

Ansvar

En grundpelare som finns i både Lean Software Development och Scrum (eller i all *agil*-utveckling) är det stora ansvaret som läggs på utvecklingsteam[1,2]. De gamla metoderna när en eller flera chefer talade om för utvecklarna vad de skulle göra är passé. Istället är det dessa personer som kommunicerar med kunden och tar fram de problem och önskemål kunden har, därefter förmedlas detta till utvecklingsteamet. Ansvaret ligger då på utvecklingsteamet att leverera bra och kvalitativa lösningar.

Detta medför att slutprodukten oftast blir bättre då utvecklingsteamet får arbeta så som de själva vill och göra det de tror är bäst. För att åstadkomma detta krävs mycket tillit och respekt gentemot sina medarbetare, och just det här genomsyrar både Lean Software Development och *agil*-utveckling.

1 Källor

1. <https://www.scrum.org/Portals/0/Documents/Scrum%20Guides/2013/Scrum-Guide-SE.pdf#zoom=100>
2. http://msdn.microsoft.com/en-us/library/hh533841.aspx#BKMK_intro
3. [http://en.wikipedia.org/wiki/Scrum_\(rugby\)](http://en.wikipedia.org/wiki/Scrum_(rugby))
4. <http://blog.goyello.com/2010/03/03/is-lean-software-development-better-than-scrum/>
5. <http://www.processexcellencenetwork.com/lean-six-sigma-business-transformation/articles/using-lean-in-agile-software-development-a-compari/>
6. https://www.ibm.com/developerworks/community/blogs/ambler/entry/principles_lean_software_development?lang=en