Plocka godbitarna ur det agila smörgåsbordet

Agila, eller lättrörliga, metoder vinner alltmer fotfäste inom systemutveckling. DSDM respektive Scrum är två av de vanligaste metoderna. Vad skiljer dessa metoder från mer traditionella metoder, och hur skiljer sig metoderna från varandra? Kan man rentav använda en mix av båda metoderna?

Det som kännetecknar en agil metod är att den skriver under på det sk Agile Manifesto som myntades av ett antal representanter för lättrörlig utveckling för sisådär en sju-åtta år sedan. Detta innebär kortfattat att man ska prioritera:

- Människor och samarbete framför processer och verktyg
- Fungerande mjukvara över mängder av dokumentation
- Samarbete med kund istället för kontraktsförhandlingar
- Lyhördhet inför förändringar snarare än att följa en plan

Därmed inte sagt att processer, verktyg och dokumentation **inte** är av betydelse. Dessa teser kan tyckas självklara i sin enkelhet, men efterlevs i själva verket sällan inom traditionella metoder och utvecklingsprojekt.

Unified Process gav oss iterativ och inkrementell utveckling, samt i viss mån riskminimering, till skillnad från vattenfallsmetoderna. Dessa egenskaper är också mycket viktiga inom de agila metoderna. Men vad är då skillnaden?

Inom agil utveckling planerar man inte aktivitetsbaserat, utan fokuserar istället på leverabler. Den viktigaste leverablen är ”fungerande mjukvara”, men det kan också röra sig om andra arbetsprodukter. Vän av ordning menar självklart att även t ex RUP har leverabler i form av dess artefakter. Skillnaden är att de agila metoderna inte på långt när har lika många. Här skiljer sig också DSDM från Scrum, där DSDM definierar en rikligare flora av roller och produkter medan Scrum egentligen bara har tre roller och ett fåtal fördefinierade produkter.

Den viktigaste skillnaden är emellertid att de agila metoderna tar riskminimering en nivå längre. Detta genom att man uteslutande fokuserar på timeboxing, vilket innebär att man alltid levererar på utsatt tid och med ett fastställt antal resurser. Istället låter man antalet funktioner vara variabel. Man tilldelar alltså inte heller resurser till aktiviteter.

När man inte exakt kan fastställa hur många funktioner som kommer att realiseras blir det oerhört viktigt att kunna prioritera krav. Det är också viktigt att användare, beställare och utvecklingsteam är involverade i detta prioriteringsarbete. Detta är också något som de agila metoderna adresserar.

Här har dock Scrum och DSDM ett något divergerande angreppsätt. Inom Scrum listas funktionerna i en sk product backlog där helt enkelt de högst prioriterade kravet ligger överst i listan. De högst prioriterade funktionerna som bedöms hinnas realiseras inom en utvecklingscykel, en sprint (t ex 30 dagar), tas med till den sk sprint-planeringen för att detaljplaneras av utvecklingsteamet. Inom DSDM arbetar man istället med sk MoSCoW-

prioritering. Detta innebär att kraven delas in i “Must-haves”, “Should-haves”, “Could-haves”, samt “Won’t have (this time)”. En timebox, analogt med sprint i Scrum, får aldrig innehålla mer än 60% “must-have” krav. Detta innebär i förlängningen även att hela projektet endast innehåller 60% ”must-have” krav. Traditionella ”skall”-krav består av både ”must haves” och ”should haves”. Det gäller mao att vara kritisk när ett krav tilldelas ”must-have”. Ett ”must-have” krav som inte realiseras sägs stjälpa hela projektet. Detta innebär å andra sidan inte att man endast levererar ”must-haves”. Men det ger beställaren incitament att verkligen fokusera på vad som är viktigt.

Ett vanligt missförstånd är att man inte skulle lägga tid på planering inom agila metoder. En av grundvärderingarna ovan är i och för sig att vara lyhörd inför förändringar snarare än att följa en plan, men det är inte detsamma som att inte planera alls. I själva verket planerar man väldigt mycket inom agila metoder. Skillnaderna är att man istället för att planera mycket i början av ett projekt, över en framtid som man inte vet något om, planerar ofta med en kortare horisont.

Vad gäller planering på detaljnivå har Scrum en styrka med den ovan nämnda sprint-planeringen. Vid denna bryter utvecklingsteamet ned kraven till funktioner om högst 16 timmars leverabler. Poängen är att ingen vet bättre hur lång en uppgift tar att göra, än den som ska utföra den.

Hur vet man då vilken metod man ska välja? Skillnaderna är tämligen små, och metoderna delar de övergripande värderingarna. Ytterligare en likhet är att ingen av metoderna talar om i detalj hur du ska bedriva utvecklingsarbetet inom ramen för en timebox eller sprint. DSDM är en mer heltäckande utvecklingsmetod och kan sägas passa bättre i en traditionell projektorganisation. En organisation med definierade produktlinjer och en någorlunda homogen utvecklingsmiljö kanske istället drar större nytta av Scrum. Behovet av definierade roller är kanske inte så stort. Man kan även dra nytta av den mer sekventiella hanteringen av prioriterade krav, för att varje sprint sedan levererar ett nytt inkrement av färdig programvara.

Sanningen är kanske att man i de flesta organisationer kan plocka godbitarna från flertalet agila metoder. Har man behov av ett antal fördefinierade roller och arbetsprodukter i en projektorganisation kanske DSDM är rätt som övergripande metod. Denna kan mycket väl kombineras med Scrum’s mer sammansvetsade utvecklingsteam och fokusering på detaljplanering inom ramen för varje timebox eller sprint. I det dagliga utvecklingsarbetet har sk testdriven utveckling visat sig höja kvaliteten på mjukvara och därmed också förvaltningsbarhet. Låt oss vara hängivna de övergripande agila värderingarna och nyttja vad de olika metoderna och teknikerna har att erbjuda.

Anders Larsson, Modul1
Styrelsen DSDM Consortium