

## Agil projektledning med Scrum: tekniker för mjukvaruutveckling - 3 dagar

*kurser 918*

- Du får lära dig att**
- Utveckla komplexa mjukvarusystem och produkter av hög kvalitet på kort tid med hjälp av agila metoder
  - Minimera osäkerhet och risk genom att tillämpa agila värderingar och principer
  - Tillämpa Scrumramverket för att möta verksamhetens operationella behov
  - Fokusera på kontinuerlig förbättring med hjälp av inbäddad inläring på arbetsplatsen
  - Införa tekniker för att tillämpa Scrum storskaliga och geografiskt distribuerade projekt
  - Efterhand ändra kulturen inom verksamheten för att dra nytta av agila metoder
- Sammanfattning** Traditionell mjukvaruutveckling kan vara en besvärlig process som präglas av osäkerhet och risk. Agila metoder och Scrumramverket tar itu med det här problemet genom att främja öppet samarbete och anpassningsförmåga under projektets hela livscykel. Dessa metoder inriktar sig på att utveckla bättre mjukvara på kortare tid. På kursen lär du dig att använda agila och Scrumtekniker för att hantera mjukvaruutvecklingsprojekt.
- Vem bör delta** Projektledare, mjukvaruarkitekter, systemanalytiker, teamledare, utvecklare och alla som vill använda Scrummetoder och agila metoder.
- RealityPlus™** Kursens fallstudier och övningar ger en djupdykning i en verklighetstrogen agil projektledningsmiljö. Övningarna inkluderar:
- Identifiera kandidater till produktägarrollen
  - Simulering av ett tre-iterationsprojekt
  - Bedöma storlek på produktfunktionerna
  - Välja funktioner att implementera i en iteration
  - Köra ett dagligt Scrum-möte
  - Planera ett projekt med flera team
  - Tolka kvantitativa data för att utvärdera projektets framsteg
  - Snabba på inläringen med verktyg och tekniker för tillbakablickande
  - Lösningar för ScrumMaster-utmaningar

## Agil projektledning med Scrum: tekniker för mjukvaruutveckling - 3 dagar

*kurser 918*

### Introduktion till agil projektledning

- Formulera agila värderingar och principer
- Jämföra agila metoder med traditionella metoder
- Agil mjukvaruutveckling: myt och verklighet
- Lokalisera Scrum i empirisk processteori

### Grunderna i Scrum för att hantera osäkerhet och risk

#### Identifiera roller och ansvarsområden

- Prioritera krav och driva projektet med hjälp av produktens ägare
- Etablera Scrum Master som en helt ny ledarroll
- Forma ett självgående Feature Team
- Kommunicera med externa intressenter

### Hantera leverans av produkter

- Skapa en produktorderstock som en lista på krav och tekniska frågor
- Iterera utvecklingen i cykler om en månad eller mindre
- Tidsramar för att skapa en regelbunden arbetsrytm

### Verktyg för att hålla koll på och övervaka ett projekt

- Återge krav som användarberättelser
- Fastställa prestationsmål med hjälp av teamets snabbhet
- Dela upp återstående arbete med burn down charts och parking lot diagrams
- Hantera utvecklingsuppgifter med task boards

### Planera ett agilt projekt

#### Fokusera på affärsvärde

- Leverera affärsvärderade funktioner med Release Planning och Sprint Planning
- Samarbeta med kunder för att hantera risker och osäkerhet
- Visa på implementerade funktioner som "potentiellt leveransklara tillägg"

### Identifiera funktioner för utveckling i en iteration

- Bryta ner stora krav i testbara och mätbara funktioner
- Konkretisera ett Sprint Goal med hjälp av val av funktioner
- Identifiera utvecklingsuppgifter i Sprint Backlog

### Främja eget initiativtagande i utvecklingsteamet

#### Skapa en optimal arbetsmiljö

- Rekrytera till funktionsteamet

- Hantera distribuerade team
- Skydda utvecklingsteamet från yttre påverkan
- Synliggöra projektets framsteg

### Övergång till eget initiativtagande

- Hjälpa teammedlemmarna att organisera sig själva
- Coacha teamet att kontrollera sin egen utvecklingsprocess

### Köra iterationer

- Mikro-planering med dagliga Scrum-möten
- Producera en kvalitetssäkrad produkt med affärsvärde
- Bestämma vad det innebär att vara "färdig"

### Hantera förändringar

#### Generera snabb feedback med Sprint Review

- Visa färdig funktionaliteten för kunden
- Vårda samarbetet med intressenter genom en diskussion om vad "färdig" innebär
- Revidera estimeringar baserat på erfarenhet

### Granska itereringar genom Sprint Retrospective

- Tänka igenom vilka lärdomar du dragit
- Integrera nya kunskaper för att uppnå kontinuerlig förbättring

### Tillämpa ett agilt arbetssätt i hela organisationen

#### Anpassa till storskaliga projekt

- Arbeta med stora produktorderstockar
- Anpassa produktägarens roll
- Koordinera funktionsteam och komponentteam

### Hantera distribuerad utveckling

- Release- och Sprint-planering för utspridda team
- Säkra effektiv kommunikation över flera tidszoner
- Hålla distribuerade Sprint Reviews och Sprint Retrospectives

### Skapa ett tempo för att med framgång börja använda agila metoder

- Bedöma organisationens beredskap för att införa agila metoder
- Skapa en Improvement Backlog för kontinuerliga förbättringar