



EXEMPEL

Utvecklingsprocess

Framtagning av applikation enligt kunds önskemål

Vi är förespråkare av konceptet "Minimum Viable Product" (MVP). En enkel översättning av betydelsen skulle kunna vara "Produkt med minsta möjliga funktionalitet". Det finns ett stort mervärde med att i första versionen skala bort så mycket som möjligt och lansera en produkt som enbart har basfunktionerna. Den måste naturligtvis fortfarande vara användbar och tillföra nytta. Nästan alltid är det bara en avskalad version av det man vill ha.

Varför förespråkar vi en sådan process? Jo, nuförtiden sker nästan all utveckling "iterativt" (förr i tiden skulle man "lite i taget"). Det finns två mycket tungt vägande fördelar med detta:

- + **Man får nytta av sin investering tidigare**
- + **Man får återkoppling från de verkliga användarna av produkten**

Oavsett om man väljer att göra en MVP eller lansera all funktionalitet på en gång så är det nu dags att börja utveckla. Det första man skapar är en lista över samtliga funktioner som systemet skall ha.

Även här är det fråga om två syften:

- **Dels behöver man en förteckning över funktionerna för att kunna skapa arkitekturen**
- **Men det är också en kontrollfunktion; i slutändan är det ett underlag för att verifiera att allt verkligen kom med.**



Funktionslista och arkitektur

Detta kan tyvärr vara ett ganska stort och tråkigt dokument, men som beställare skall man tänka på att man inte får det man vill ha utan det man beställer, och om dom två sakerna inte är samma så är det risk att man blir besviken.

Tack och lov finns det en mängd kontrollstationer på vägen där man har möjlighet att ändra sig, men om man siktar fel från början så brukar det medföra kostnader och eventuellt förseningar.

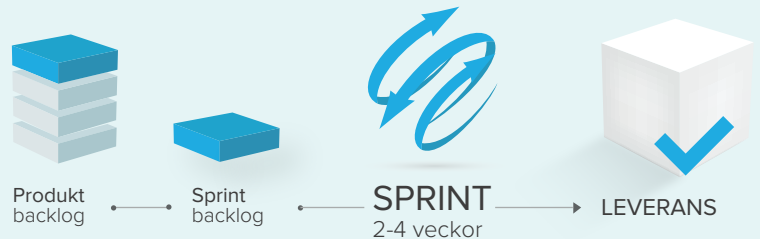
Precis som med moderna bilar så sätts dagens programvara ofta ihop av delar. Vissa av dessa är standardiserade eller ingår i något system som man baserar sin lösning på. Andra delar (oftast de som är unika för produkten) måste skapas speciellt för applikationen. Arkitekturens roll är att bestämma vilka delar som behövs för att bygga applikationen, vad som måste utvecklas själva och vilka standard-komponenter man kan använda. Dessutom måste man beskriva hur delarna passar ihop och hur de pratar med varandra.

Det gäller inte bara att få med alla funktioner; prestandan måste vara tillräcklig, systemet måste vara säkert och det måste gå att underhålla.



Utveckling och test

När allt ovanstående är klart ska systemet skapas. Normalt jobbar vi i etapper ("Sprintar").



Efter varje "sprint" visas det som är utvecklat. Denna kontrollpunkt är dels för att beställaren ska kunna bilda sig en uppfattning om hur jobbet framskrider, men den har också en ytterligare viktig funktion: Det är oftast först när man får se och känna på något som man riktigt förstår hur det borde vara; ofta finner man detaljer som behöver ändras eller funktioner som man missat. Och ju tidigare man hittar missarna desto billigare är det att ändra sig. En etapp är ofta 2-4 veckor lång.

Det mesta av utvecklingsarbetet sker utan att beställaren måste följa det dagliga arbetet, men då och då dyker det upp frågor som måste besvaras. Dessa frågor får man ta på ett informellt sätt. Vi använder ett Webb-baserat verktyg för att spåra varje moment. De delar av programvaran som utgör användargränssnitt kräver generellt mer återkoppling från beställaren, eftersom det inte bara är en teknisk lösning utan också i stor utsträckning en fråga om tycke och smak. I dessa delar kan man behöva ha kontinuerliga möten.



Driftsättning och förvaltning

Nu börjar förvaltningsaktiviteterna. Systemet skall övervakas, säkerhetskopior tas, prestanda mätas, fel rapporteras etc. Så småningom har man kanske samlat på sig en lista av förändringar som skall genomföras och även en annan lista med fel som upptäckts. Utifrån dessa brukar man genomföra projekt som syftar till nya versioner av programvaran. I modernare lösningar har man istället infört en kontinuerlig process med många små uppdateringar.

Förstudie EXEMPEL

- Omvärldsanalys av marknaden för applikationen som finns i åtanke.
- Val av teknisk lösning för tjänsten
- Prioriteringar gällande tid, kostnad och resultat.
- Produktkrav och projektkrav.

Omfattning: 160-200/h

Ansvarig: Teknisk projektledare

Deltagare: Teknisk projektledare, projektledare.

Startdatum:20180903

Slutdatum:20181001

Förväntat resultat:

- Vad finns att tillgå idag.
- Produktkrav.
- Projektplan.

Arkitektur samt kravinsamling, kravställning

Skapa en arkitektur för att hantera säker digital kommunikation i applikation samt samla in krav till kravställningen. Kostnaden för att åtgärda fel i IT-system ökar exponentiellt ju senare i skedet felen upptäcks, ända upp till 100 gånger dyrare att åtgärda i produktion än i kravställning.

Omfattning: 400-600/h

Ansvarig: Arkitekt, Projektledare

Deltagare: Arkitekt, projektledare och scrum master

Startdatum:20181002

Slutdatum:20181015

Förväntat resultat: Följande skall vara framtaget:

- Datamodell.
- Systemkarta.
- Processkarta.
- Funktionella krav.

Utveckling och test

Utifrån Arkitektur/Krav sammanställning utveckla applikation för säker digital kommunikation. All utveckling sker i 2 veckors sprintar enligt SCRUM metoden för att vara så agila som möjligt. Test sker kontinuerligt i sprintarna. Acceptans tester sker efter utvecklings periodens slut.

Omfattning: 2000-4000/h

Ansvarig: Teknisk projektledare, Projektledare.

Deltagare: Utvecklare 1, Grafisk designer, Konsult 1, Testare 1, Scrum master, Projektledare, Teknisk projektledare.

Startdatum:20181016

Slutdatum:20190131

Förväntat resultat: En stabil version 1 av applikation. Programvara som är skalbar och motsvarar kravställningarna.

Deploy - Installationspaket

Framtagning av installationspaket till applikation.

Omfattning: 80-100/h

Ansvarig: Teknisk projektledare.

Deltagare: Deploy ansvarig, Teknisk projektledare.

Startdatum:20190201

Slutdatum:20190215

Förväntat resultat: Installations paket som personer kan ladda ner och installera.