



Guide:

Att utveckla en **egen** affärsapplikation

Processen där behovet styr produkten

Att utveckla en egen applikation

Att utveckla en egen applikation kan framstå som både svårt och skrämmande. En vanlig anledning till detta är att den slutgiltiga produkten inte är helt känd och vägen dit upplevs som krokig och oklar. Samtidigt är flexibiliteten i denna process en stor del av värdet, en process där behovet helt styr produkten. Denna guide beskriver viktiga delar av denna process på ett övergripande sätt, med förhoppningen att den ska kännas en aning mindre skrämmande.

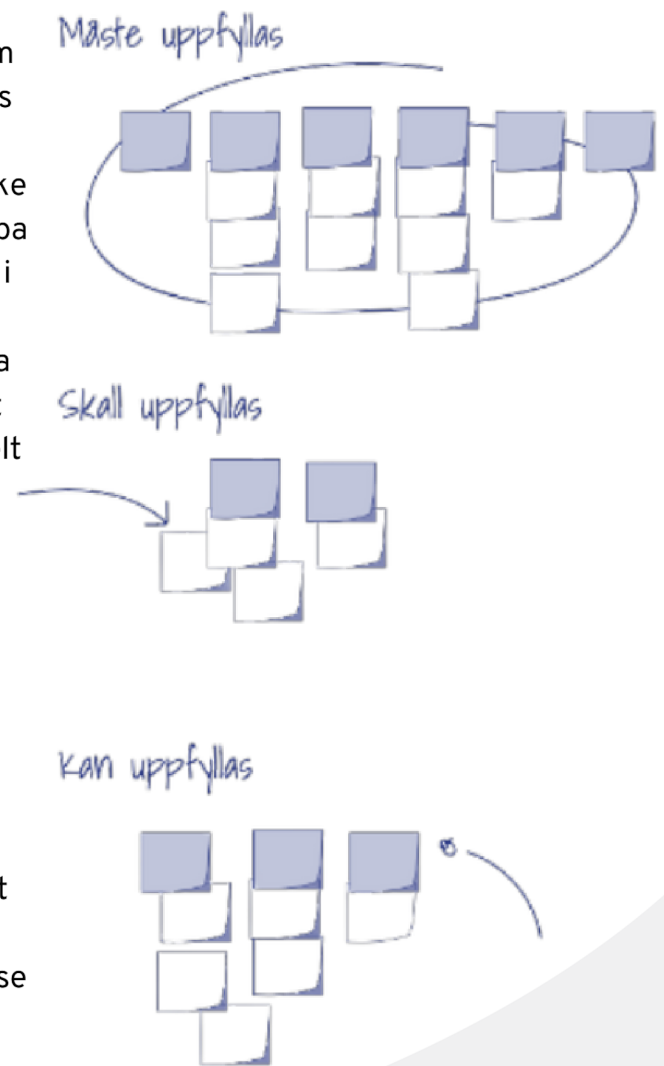


Förstå behovet och skapa en gemensam bild av helheten

En väldigt central del av att utveckla en affärsapplikation handlar om att etablera en gemensam bild av de behov som ligger till grund för applikationens existensberättigande. Det finns i detta läge flera olika perspektiv som är viktiga att ta i beaktande, men kanske viktigast är det affärsvärde som applikationen ska hjälpa till att uppfylla. Andra perspektiv som också måste tas i beaktande är slutanvändarens och den mottagande organisationens. Alla dessa perspektiv på den tilltänkta applikationen i sin verkliga miljö ger därefter möjlighet att definiera ett antal olika effektmål som gör det enkelt att i efterhand följa upp utfall mot den initiala målsättningen.

Exempel på effektmål: “Tiden som organisationen lägger på inventering ska minskas med 25 %.”

För att komma fram till en förståelse för hur applikationen används i sin verkliga miljö är det viktigt att genomföra en workshop. Ett format som är vanligt och som fungerar bra är story mapping. Det innebär att man med utgångspunkt i målsättningen och de olika perspektiven gemensamt kommer fram till en berättelse om applikationens verklighet. På detta sätt kan behov och målsättningar enkelt brytas ner i mer konkreta så kallade User Stories som sätter organisation, slutanvändare och behov i fokus. Med ett sådant fokus blir det därefter enklare att prioritera mellan dessa olika behov eftersom kopplingen till affärsvärde blir tydlig.



Vad är user stories?

En User Story är ett sätt att formulera önskemål om funktionalitet för ett system. Framvuxet i agila metoder där kraven förutsätts vara föränderliga adresserar man problemen med den traditionella kravspecifikationen.

User Stories kompletterar det traditionella kravets “vad?” med “vem?” och “varför?”. Tanken med User Stories är att de ska vara kortfattade och kunna skrivas av såväl beställare som användare och utvecklare.

Det finns lite olika sätt att strukturera en User Story på. **Gemensamt är att alla sätt utgår från en användares perspektiv. Exempelvis: “Som säljare kan jag ändra status på en offert så att jag vet vilket skede den är i”.** Tidigt i ett projekt är en User Story ofta mindre detaljerad: “Som säljare kan jag hantera mina offerter”, och kallas då för en Epic. Efterhand bryts dessa ner och blir mer detaljerade. I de flesta fall bör de även kompletteras med information kring hur implementationen ska ske.

En User Story går snabbt att skriva och uppmuntrar till diskussion. Den korta formuleringen till och med förutsätter dialog om vad som avses med den. Att dialog behövs är allt som oftast sant också för den mest välformulerade av kravspecifikationer. Med User Stories blir kravarbetet mer transparent och det gör förväntningarna mer balanserade. Eftersom man också vet för vem funktionen ska finnas och varför så underlättas eventuell framtida förändring och prioritering.

Exakt hur funktionaliteten ett system ska ha formuleras kan tyckas vara en mindre viktig detalj – möjligtvis stämmer detta. Att löpande diskutera, prioritera och förändra funktionaliteten är desto viktigare. Syftet med User Stories är att främja detta.



Definiera lösningen **löpande** och tillsammans

Efter att ha gjort en första nedbrytning av behovet finns en gemensam förståelse för vad applikationen ska lösa. Däremot finns det mängder av detaljfrågor kvar att reda ut. Prioriteringen från Story Mapping-övningen guidar projektet i vad som bör redas ut först.

De User Stories som skrivits behöver i vissa fall brytas ner ytterligare. En Story som lyder “Som säljare vill jag kunna se en lista med kunder” bryts ner till “Som säljare vill jag kunna se en lista med de kunder som jag sålt till senaste halvåret” och “Som säljare vill jag kunna se en lista med mina A-klassade kunder”.

För att skapa en ännu tydligare bild av den tänkta lösningen kan det i många fall vara bra att arbeta med wireframes eller mockups. På så sätt kan enskilda User Stories eller större delar av den tänkta applikationen visualiseras innan utvecklingen av de startar.



Agil utveckling för att tidigt leverera värde

Den agila utvecklingsmetodiken uppkom ur problematiken i att definiera hela lösningen innan utveckling påbörjades. Detta orsakade en stor initial kostnad för en dokumentation som med största sannolikhet skulle behöva genomgå stora förändringar under utvecklingen, som följd av förändrade förutsättningar och nya upptäckter längs projektets gång.

Genom att tidigt påbörja utvecklingen av den prioriterade och definierade funktionaliteten går det att snabbt leverera en fungerande version av applikationen. Med ett fokus på det som verkligen måste finnas för att uppfylla behovet kan applikationen tidigt börja leverera ett värde. Detta är ofta inte en version som man är helt nöjd med, men en applikation som stöttar bättre än det arbetssätt som fanns tidigare. Det behöver alltså inte vara slutet på projektet, utan bara en första lanserad version av applikationen. Detta möjliggör också snabb feedback från beställare och användare samt ger möjligheten att kontinuerligt styra utvecklingen dit den ger mest affärsvärde.



Kontinuerlig testning under projektet

En mycket central del i en agil utvecklingsmetodik är att löpande och kontinuerligt testa och ge feedback på de funktioner som utvecklas. När testning sker löpande ges goda förutsättningar för att tidigt upptäcka missförstånd och rätta till dessa med minsta möjliga insats. Precis som att det är viktigt att tidigt börja med utveckling av applikationen är det därför även viktigt att tidigt börja med tester. Då det finns en gemensam bild av allt från affärsvärde till funktioner blir det även mycket enklare att tillsammans genomföra de tester som krävs för att lösningen i slutändan ska kunna accepteras för driftsättning. Den gemensamma bilden gör att både mottagare av applikationen och utvecklare av densamma förstår vilka tester som är viktiga och varför.



Nå ut till användarna

Den viktigaste faktorn i att nå ut till användarna av applikationen är att involvera dem tidigt i processen. Att utse ett antal Super Users som är involverade under hela resan från behov till testning bidrar ofta väldigt positivt till den acceptans som uppnås vid lansering. Dels så har dessa Super Users kunnat bidra med kunskap från den verklighet där applikationen ska användas och dels kan de fungera som ambassadörer för applikationen ute i verksamheten.



På Exsitec har vi stor erfarenhet av att utveckla moderna affärsapplikationer utefter våra kunders behov. Vi har levererat allt ifrån mindre applikationer för exempelvis tidrapportering och orderläggning till stora komplexa verksamhetssystem.

Funderar du på hur ni kan **ligga steget före i er bransch? Kontakta oss för rådgivning.**

hej@exsitec.se | 010-333 33 00