

VERKSAMHETSARKITEKTUR OCH AGIL SYSTEMUTVECKLING

Samarbetet runt Product Backlog



Författare

Martin Insulander

martin.insulander@irm.se

Eric Quist

eric.quist@irm.se

Magnus Elfström

magnus.elfstrom@irm.se

Publicerad

Oktober 2011

Version

1.0

Versionshistorik			
Ver	Datum	Utfärdare	Kommentar
1.0	2011-10-25		Första publika versionen

Inledning

Detta är en övergripande metodbeskrivning för hur man går från verksamhetens processer, via processmodellering och informationsmodellering, till en backlog (prioriterad lista med arbetspunkter) som är anpassad för iterativa, lättrörliga utvecklingsmetoder. I det här fallet har vi använt Scrum som exempel, men det fungerar på liknande sätt med t.ex. Kanban.

Beskrivningen är tänkt för dig som jobbar med verksamhetsarkitektur (VA) och vill hitta ett sätt att kommunicera med agila utvecklingsteam på "deras språk", eller vice versa, för dig som jobbar med Scrum eller liknande metod och som vill lära dig mer om verksamhetsarkitekters verktyg.

Många som arbetat med Scrum upplever att det är just produktägarens roll som är svårast. Kommunikationen mellan utvecklare och produktägare blir ofta en flaskhals, trots att den är oerhört viktig och dessutom behöver vara dubbelriktad för att prioriteringen verkligen ska fungera. Arbetspunkterna blir lätt otydliga eller ofullständiga och missförstånd mellan utvecklarna och produktägaren uppstår. Många organisationer tillsätter produktägare från verksamhetssidan men allokerar inte tillräckligt med tid. Andra lägger rollen på någon på IT-sidan och riskerar då att sakna personer med verksamhetsperspektiv i utvecklingsarbetet.

Produktägarens uppgift är att kontinuerligt arbeta med kravställning, vilket helst bör ske på ett strukturerat sätt (och låta det ta tid). Scrum (i Jeff Sutherlands och Ken Schwabers originalversion) innehåller ingen metodik för kravfångst – att ta fram punkterna i backlogen. Den här metoden kan förhoppningsvis ge just den strukturen, åtminstone i projektets början, genom att återanvända IRM:s beprövade metoder för affärsutveckling, verksamhetsarkitektur och kravställning.

TIPS

Vår erfarenhet är att många verksamheter som arbetar med verksamhetsarkitektur och som vill utveckla enligt agila metoder inte kommer framåt i hur detta ska fungera tillsammans. Vi rekommenderar då att man börjar med att komma överens om vad man behöver kunskap om för att lyckas med sin utveckling och i nästa steg arbetar igenom hur, dvs processen för arbetet. På så vis undviker man diskussioner med meningslösheter som "det där är vattenfall" eller "fort, men fel är inte bra".

Bakgrund

Här följer ett antal orienterande avsnitt som beskriver Scrum och verksamhetsarkitektur mycket kortfattat. Om du som läsare känner dig välbekant med båda områdena kan du hoppa direkt till Metodbeskrivningen nedan.

Scrum

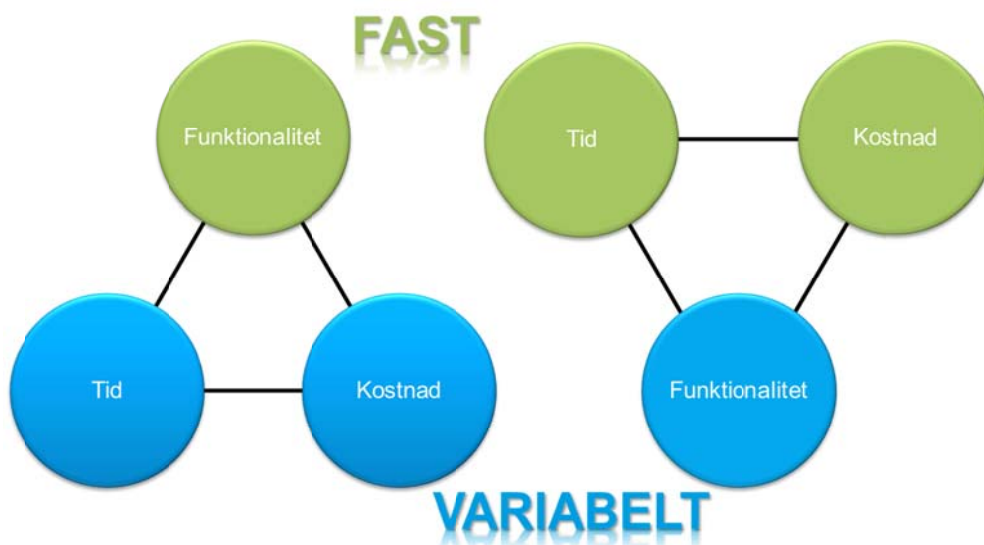
Lättrörliga (agila) utvecklingsmetoder, framförallt Scrum, har fått ett stort uppsving inom IT-branschen de senaste åren. De flesta som lyckas införa agila metoder upplever att man lyckas leverera högre kvalitet mer effektivt och dessutom ha en trevligare arbetssituation under processens gång.

I Scrum definieras tre roller:

1. Scrum Team – utvecklarna, oftast 3-7 st
2. Scrum Master – ansvarar för att processen i sig fungerar
3. Produktägaren (eng. Product Owner) – prioriterar och är kundens "kommunikationskanal"

Produktägaren är den som ansvarar för kravarbete. Skillnaden mot traditionella metoder (vattenfall) är att det oftast inte finns någon kravspecifikation som täcker hela produkten från början till slut, utan en backlog. Utformningen på backlog definieras inte av författarna till Scrum, men de flesta har något som kan liknas vid en kravlista. Den stora skillnaden är bara att backlogen är under ständig förändring och prioriteras, detaljeras och diskuteras under hela utvecklingsarbetets livscykel.

Scrum som projektstyrningsmetod bygger på att man istället för en fastställd funktionalitet har satt en fast kalendertid och kostnad (resurserna) för en iteration (sprint). Sprinten har ett tydligt mål där man planerar in att ett antal användarberättelser (user stories, från Extreme Programming (XP)) som ska färdigställas. Eftersom tiden och kostnaden är fast så innebär det dock att det inte är säkert att alla användarberättelser kommer att bli klara. Anledningen till att man gör så här är att funktionaliteten inte är känd i förväg och därför mycket svårare att ha kontroll över än tid och kostnad.



Backlog

Product Backlog är kravspecifikationen i Scrum. Det är en prioriterad lista på önskad funktionalitet.

Vad ska backlogen innehålla?

Om detta tvista de lärda. Det viktigaste är naturligtvis att detaljnivån gör att utvecklingsteamet och produktägare kan kommunicera på ett bra sätt.

Vissa anser att product backlog enbart ska innehålla krav från verksamheten, dvs inga tekniska arbetspunkter. Detta ger en mer renodlad backlog där alla punkter (förhoppningsvis) har direkt koppling till verksamhetsnytta. Å andra sidan måste tekniska och lösningsinterna aspekter av arbetet då kommuniceras på något annat sätt.

En annan variant är att backlogen i första hand fylls med krav från verksamheten, men att tekniska bidrag också förs in allt efter som. Detta kan medföra att det blir väldigt många punkter, så viss filtrering eller sammanslagning kan vara lämplig. En fördel med denna metod är att man kan beakta tekniska förutsättningar och tidsuppskattningar för dess implementation när man prioriterar backlogen. En annan fördel kan vara att utvecklingsteamet blir mer involverat – det blir ett mer dubbelriktat flöde till och från backlogen.

En tredje, vanligt förekommande variant är att product backlog endast innehåller verksamhetskrav, men att man vid varje utvecklingsiterations början för in tekniska bidrag och förutsättningar. Detta kan ge "det bästa av två världar" men kräver att produktägaren kan göra snabba prioriteringar, oftast inom tidsramen av ett planeringsmöte.

Backlogens format

De flesta väljer att skriva punkterna i backlogen som användarberättelser.



Användarberättelser följer mallen:

Som [roll] vill jag [mål/önskemål] så att [resultat/nytta].

Exempel:

Som kund vill jag se mina beställningar på förstasidan så att jag kan få en snabb överblick utan att navigera vidare.

Denna mall fungerar ofta bra och gör det lätt för kravställare och utvecklare att förstå varandra. Den ska dock inte följas slaviskt. Exempelvis passar användarberättelser inte särskilt bra till tekniska arbetspunkter (som ofta tillkommer som lösningsrelaterad konsekvens för att en användarberättelse kräver vissa tekniska förutsättningar).

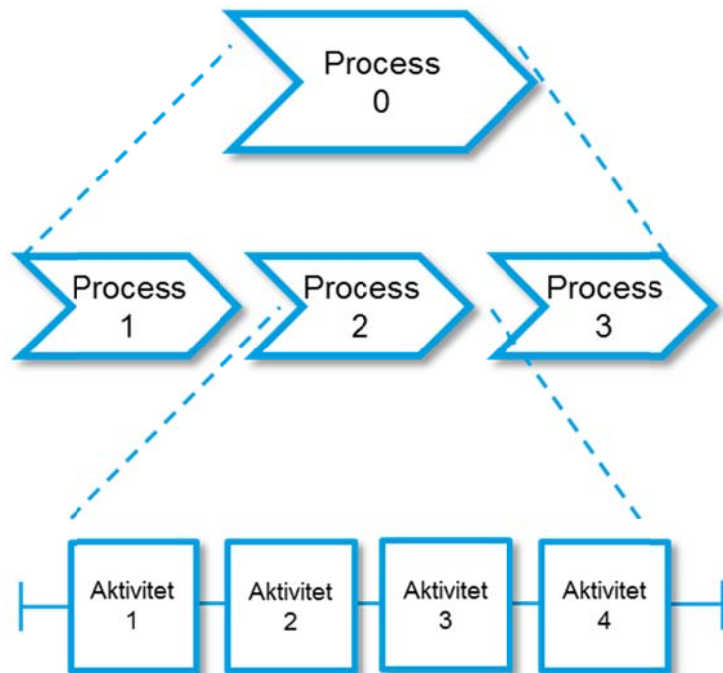
Exempel:

Sätt upp automatiska integrationstester för överföringen av kundregister.

Processer

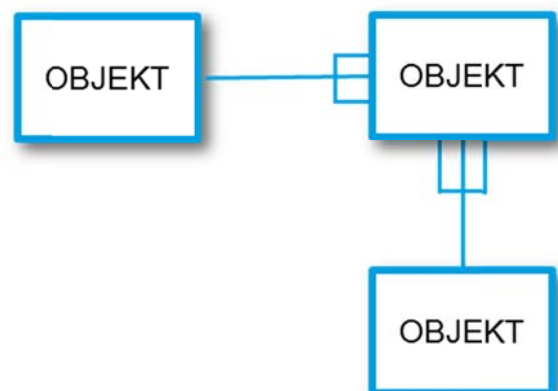
En verksamhet utvecklas idag hela tiden för att nå den vision som satts upp på lång sikt och de mål som finns på mellanlång och kort sikt. Det vanligast förekommande och mest accepterade sättet att beskriva hur en verksamhet fungerar är att använda sig av processer. En process initieras av en händelse och består av en samling aktiviteter som tillsammans skapar ett resultat som utgör ett värde för kunden. I agila metoder finns ett stort fokus på att man framför allt ska koncentrera sig på det som ger ett värde. Men metoderna för att definiera värdet är ofta

mer "lokala" och kan lätt missa det större målet för verksamheten. Ironiskt nog så misslyckas många processinitiativ då de fokuserar på att rita upp ett antal processer eller aktiviteter, men helt har glömt bort att definiera vem som är kunden som ska ta emot resultatet från processen. De som gjort det glömmar även bort att definiera vilket värde (snabbhet, kvalitet, lågt pris eller god service) som processen ska ge. Processer kan beskrivas på en övergripande nivå i en processkarta som sedan bryts ner till en nivå där en process beskrivs av ett antal aktiviteter i en processmodell. En aktivitet definieras som något som en person utför vid ett tillfälle utan något egentligt tidsglapp.



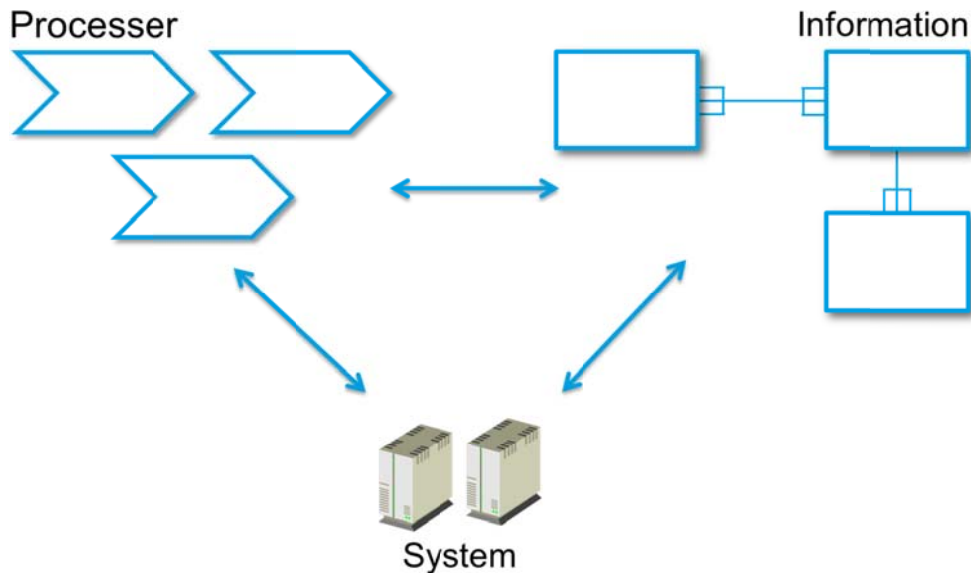
Information

Det finns inga processer som kan fungera utan information. Information är en värdefull resurs som varje verksamhet behöver vårda för att kunna ha fungerande processer som är värdeskapande för kunderna. Informationen beskrivs i rika informationsmodeller där även många verksamhetsregler fångas. Brister i kunskap, definitioner och kvalitet på informationen är ett stort problem för de flesta verksamheter idag. Problemen orsakas till stor del av att samma informationsmängd hanteras i många olika system samtidigt som det inte finns en klar ägare/källa till varje informationsmängd.



Verksamhetsarkitektur

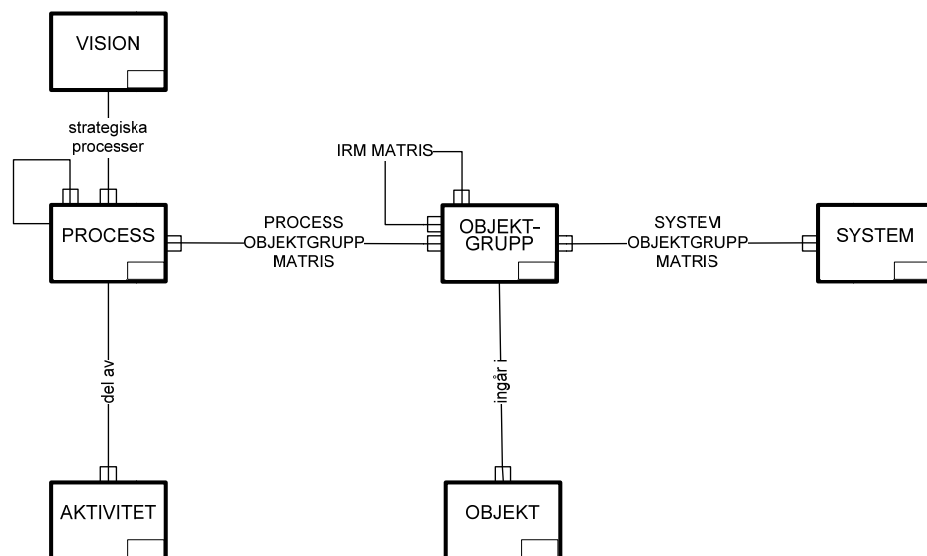
IT-systemen utgör den tredje hörnstenen i en verksamhetsarkitektur.



En första bild, en stadsplan, för verksamheten får man genom att analysera sambanden mellan dessa hörnstenar. En initial analys kan göras genom att i en matris visa på vilken information som behövs (läses) samt vilken information som skapas i vilken/vilka processer. Vidare behöver man också kartlägga i vilka system som informationen skapas respektive bara används. Vanligtvis kommer man här kunna se väldigt stora skillnader i verksamhetens önskan (process/information-matrisen) jämfört med hur stödet ser ut (system/information-matrisen).

Slutligen gör vi en ideal matris (IRM-matris) där en funktion (förmågan att vårda informationsresursen) bara skapar en informationsmängd en gång, men där den används på många ställen.

Detta kan sammanfattas i följande informationsmodell:



Stadsplanen visar på en mycket övergripande nivå hur IT-stödet till verksamheten ser ut idag och kan även omfatta idéer på vilka som är de närmaste/första viktiga stegen mot att förbättra dagens situation så att verksamheten som helhet kan bli mer agil.

Denna första idé och ögonblicksbild av verksamhetsarkitekturen är bara första steget, men arbetet med verksamhetsarkitekturen pågår kontinuerligt, genom att man både tillför och tar med sig kunskap från de enskilda projekten.

Projekt

De flesta verksamheter bedriver kontinuerlig verksamhetsutveckling i projektform. Projekten uppstår antingen strukturerat utifrån t.ex. en utvecklad affärsmodell eller mer löst utifrån ett behov som verksamheten upplever att de behöver utveckla. Framtagna processkartor kan i båda fallen användas för att göra en initial avgränsning av projektets tänkta omfattning.

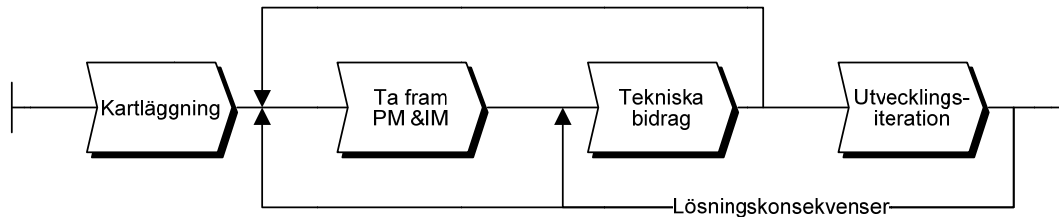
För att långsiktigt nå en agil verksamhet behöver verksamhetsarkitekturen kontinuerligt förbättras och i det avseendet måste varje projekt bidra med sin del för i annat fall kommer projektet göra situationen värre. Det innebär att projekten också behöver mätas på fler punkter än enbart tid och pengar.

I denna metodbeskrivning har vi gjort en avgränsning i att bara beskriva samarbetet runt product backlog och vi beskriver inte hur man identifierar och planerar krav- och lösningsinitiativ. Detta är naturligtvis viktigt för att få en agil verksamhet då den inte är betjänt av att flera initiativ realiserar samma eller befintliga delar i ett agilt utvecklingsinitiativ.

Metodbeskrivning

Översikt

Varje steg i processen leder till att punkter förs över in i en lista (backlog) och prioriteras. Till en början består backlogen av grova block (med ursprung i processer) och för varje steg blir detaljnivån högre längre upp i backlogen.



Den första delen av kravställningen skiljer sig inte från traditionella metoder. Här bygger man en övergripande bild av behoven och försöker täcka in hela kravbilden utan att gå ner på detaljer. Eftersom utvecklingsarbetet troligen inte kan påbörjas (ingen prioritering är gjord) förrän denna del är klar vill man hålla den så kort som möjligt.

När en övergripande kravbild (processkarta och informationsmodell) är framtagen förs processerna över på en lista (backlog) och prioriteras. Den första processen i listan (högst prioritet) specificeras i högre detaljgrad – en processmodell tas fram och vid behov kompletteras även informationsmodellen med mer detaljer. Aktiviteterna från processmodellen förs över i backlogen och prioriteras, både inbördes och gentemot de befintliga punkterna i backlogen.

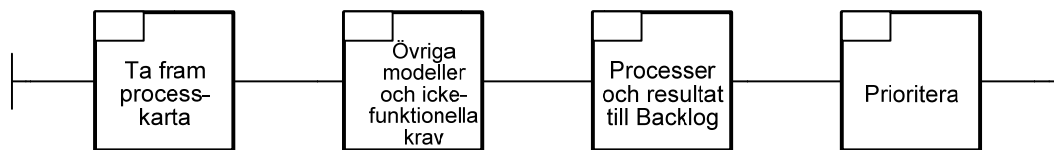
Nästa steg styrs av vad man vill att backlogen ska innehålla (se diskussionen i föregående avsnitt). För varje punkt i backlogen identifieras eventuella tekniska förutsättningar som krävs för att aktiviteten ska kunna implementeras. Detta görs i samråd med utvecklare. De nya arbetspunkterna förs över i backlog och prioriteras.

När detaljnivån är tillräcklig och det finns tillräckligt många arbetspunkter i backlog för att första iterationen i utvecklingsarbetet ska kunna starta, fortsätter sedan även kravarbetet iterativt och sker alltså parallellt med utvecklingsarbetet. Nästa process modelleras, informationsmodellen detaljeras, aktiviteter och eventuella tekniska förutsättningar förs över, prioriteras, bryts ned och tidsuppskattas för nästa sprint.

Eventuella lösningsekvenser, "lessons learned" under utvecklingsarbetet förs kontinuerligt (men förmodligen mest fokuserat efter varje utvecklingsiteration) in i backlog och kanske också reviderar processmodellen och informationsmodellen.

Det iterativa hjulet är därmed igång och snurrar tills projektet tar slut (om det gör det).

Steg 1. Kartläggning



Ett naturligt första steg i kartläggningen är att identifiera verksamhetens processer och dess resultat. Processerna är det främsta underlaget till att skapa en backlog. Vi rekommenderar att man redan i kartläggningen också gör en övergripande informationsmodell eftersom det ofta är svårt att få kvalitet i processkartan utan att man har enats om begreppen.

För verksamheter som redan jobbar mycket med verksamhetsarkitektur kan processkartor, process- och informationsmodeller redan vara framtagna, vilket naturligtvis snabbar upp processen och det räcker förmodligen med en översikt av befintligt material och avgränsning.

Beroende på karaktär på systemet och dess omvärld kan det i kartläggningsskedet vara nyttigt att ta fram övriga modeller, t.ex.

- Objekt-systemmatris – framförallt för att stödja avgränsning och beslut om integration. Detta förutsätter att en övergripande informationsmodell tas fram.
- Kritiska framgångsfaktorer – för att stödja prioritering
- Modeller över nuvarande systemstöd

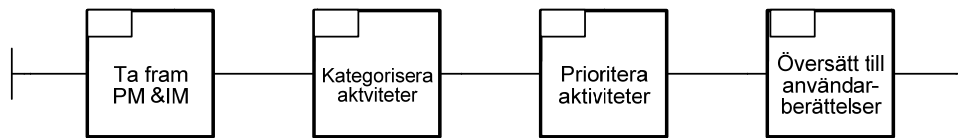
I kartläggningsskedet kan det även vara värdefullt att bilda sig en övergripande uppfattning om icke-funktionella och icke-modellrelaterade krav. Sakerna som inte kan härledas till processer kommer förmodligen inte föras in i backlog just i detta steg. Dock ger de viktig vägledning i framtida steg och bör därför dokumenteras.

När en processkarta är framtagen förs processerna och dess resultat över i en lista. Redan här kan prioritering göras, vilket lämpligtvis främst görs av verksamheten. Det kan typiskt göras efter att processkartan ritats upp i en workshop. En annan metod är att skicka ut en lista med processerna och be mottagarna poängsätta varje process på en skala 1-5, och låta totalpoängen motsvara prioritet.

Sortera listan i prioritetsordning. Högsta prioritet är högst upp i listan. Detta är den första (mycket grova) backlogen.

I agila metoder värdesätts förmågan att förändra högre än att följa en plan. Hur mycket man behöver planera i förväg varierar dock mycket med vilket problemområde man befinner sig i. Det man vill uppnå är att fatta så välinformerade beslut som möjligt och därmed så sent som möjligt, men inte senare än nödvändigt. Vad innebär det då? Om en verksamhet ska sätta upp en ny fabrik så kommer man behöva fatta fler beslut tidigt eftersom mycket är svårt eller omöjligt att ändra på. En fördel med mjukvara är att den går att ändra väldigt sent (om den är rätt konstruerad). Även i systemutveckling finns det ibland situationer där beslut behöver fattas tidigt. T.ex. om mjukvaran utgör ett samarbete mellan alla medlemsländer i EU så behöver fler beslut fattas tidigare än om man ska utveckla ett tidrapporteringsystem för en mindre konsultverksamhet.

Steg 2. Processmodell och informationsmodell



Börja med den översta processen i backloggen från steg 1. Ta fram modeller för processen och den information som används/krävs av aktiviteterna i processen. Ordningen spelar här mindre roll utan bör anpassas till situationen. Antingen görs begrepps- och informationsmodell först och processmodell därefter, eller vice versa, eller så låter man båda växa fram parallellt.

Var noga med att inte detaljera några modeller mer än vad som krävs för att kunna implementera processen och dess aktiviteter.

Kategorisera aktiviteterna i processen, t.ex. genom att färgmarkera dem. Om processen har fler resultat, identifiera om något resultat har högre prioritet än något annat. Skilj på aktiviteter som är

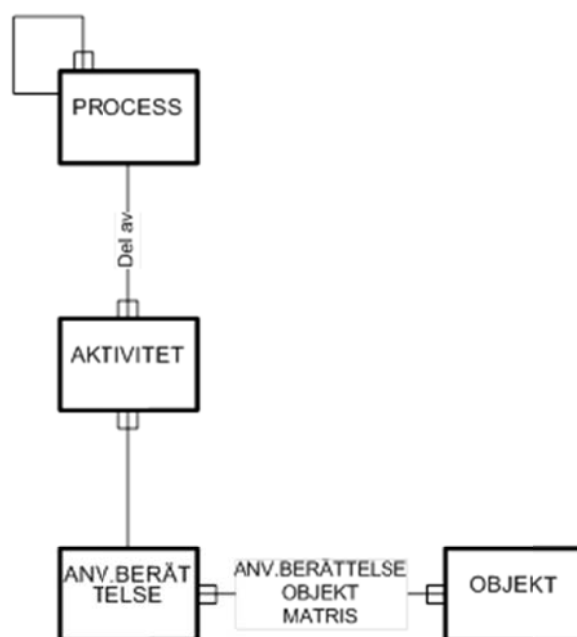
- Obligatoriska (för att kunna nå något resultat av processen)
- Icke obligatoriska (kan implementeras senare)
- "Extra features" (kan implementeras efter att hela systemet i övrigt är implementerat)
- Ej systemstöd/manuella aktiviteter/i annat system utan integration

De obligatoriska aktiviteterna prioriteras sedan inbördes och förs in i backlog. Arbetspunkterna för aktiviteterna i backloggen ersätter processens rad från steg 1.

Övriga aktiviteter förs sedan in i backlog, men prioriteras var för sig mot hela backloggen. En viss icke obligatorisk aktivitet kan mycket väl ha lägre prioritet än andra processer.

Skriv användarberättelser för de aktiviteter som förts in i backlog. När man formulerar användarberättelsen märker man ibland att processmodellen är på för grov nivå och att aktiviteterna behöver brytas ner ytterligare.

Formatet på backloggen bör innehålla kolumner för process, resultat och aktivitet, så att deras ursprung kan spåras tillbaka till modellerna. Det gör det tydligt för produktägaren i vilket sammanhang en användarberättelse hör hemma och kan på så sätt vara ett stöd i samband med t.ex. prioriteringen. För en utvecklare ger det också spårbarhet hela vägen upp till processkartan vilket gör det enklare för en utvecklare att bättre förstå vilket som är det primära kundvärdet som en process ska uppfylla till en eller flera kunder.

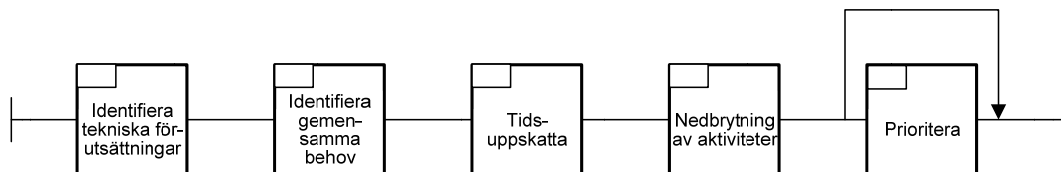


TIPS

Om du använder Visual Studio Team Foundation Server (TFS) som stöd för utvecklingsprocessen så kan "Area" användas för att peka ut processen.

Prioriteringar i detta steg görs, precis som i steg 1, i samråd med representanter för verksamheten.

Steg 3. Tekniska bidrag

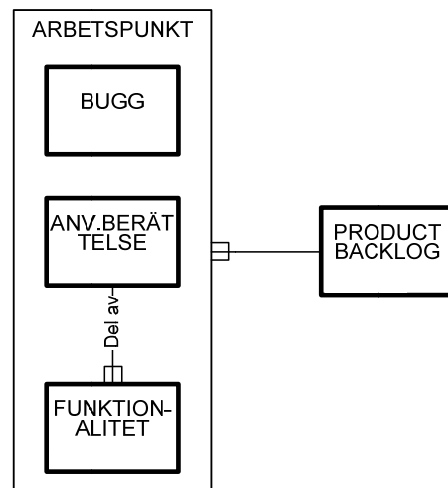


Detta steg görs i samråd mellan produktägaren och utvecklingsteamet. Om product backlog enbart ska innehålla verksamhetskrav görs detta steg som en del i planeringen inför en utvecklingsiteration. I andra fall kan det göras som en direkt följd av föregående steg.

Identifiera tekniska förutsättningar för varje användarberättelse. Kräver implementationen tekniskt förarbete som är tillräckligt omfattande för att vara en egen arbetspunkt i backlogen? För i så fall in denna som en egen punkt i backlogen ovanför.

Ett exempel kan vara att informationsmodellen framtagen i steget innan visar att det krävs mycket databasrelaterat förarbete för att implementera en viss användarberättelse.

Identifiera även gemensamma behov hos användarberättelse. Om exempelvis tre användarberättelser har gemensamt att viss funktionalitet förutsätts kan det vara bättre att bryta ut den funktionaliteten som en egen arbetspunkt (ovanför i backlogen), för att bättre kunna prioritera mellan användarberättelserna. Det uppstår då ett beroende mellan användarberättelserna (och därmed en begränsning i hur man kan prioritera) som bör markeras i backlogen.



Över tid kommer även annat att fyllas på i product backlog som t.ex. buggar.

Låt även lösningsteamet bryta ned punkter i backlogen som kräver mycket implementationsarbete i mindre delar där detta anses lämpligt.

Den översta delen av backlogen är nu i den detaljnivå som krävs för att implementationsarbetet ska kunna starta. Nu återstår bara att tidsuppskatta varje arbetspunkt. Detta görs av utvecklingsteamet.

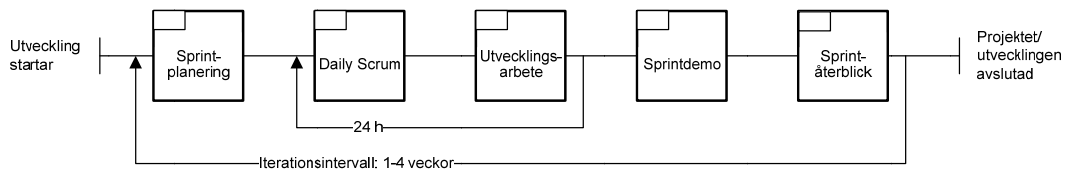
INFO:

Tidsuppskattning i Scrum-projekt använder ofta ett poängsystem där utgångspunkten är att 1 poäng motsvarar en ideal arbetsdag utan några som helst störmoment. Efter några sprintar vet teamet med allt större säkerhet hur många poäng som teamet levererar i en sprint för just det här projektet med just de här förutsättningarna. En del i det kontinuerliga förbättringsarbetet är att försöka förbättra processen så att fler poäng kan levereras i varje sprint.

Baserat på den nya informationen (tidsuppskattningen) kan det nu vara läge att omprioritera. Något som går snabbt att implementera kan exempelvis vara lägligt att tidigarelägga för att snabba upp feedbackloopen och öka moralen i största allmänhet.

Steg 4. Utvecklingsiteration

Om utvecklingsteamet använder Scrum ser processen ut ungefär som följer:

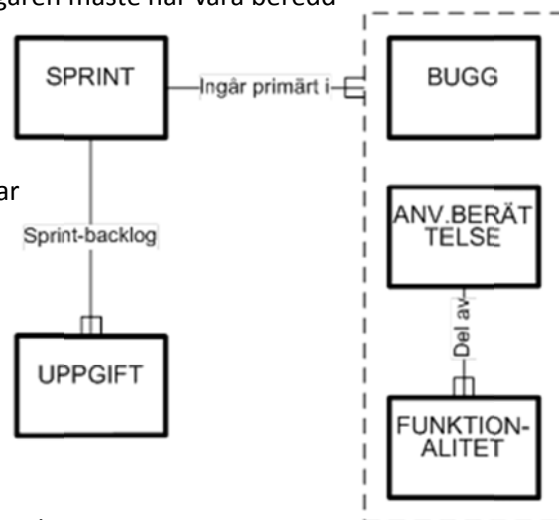


En sprint (iteration i Scrum) börjar med ett planeringsmöte där utvecklingsteamet och produktägaren förhandlar om vilken funktionalitet som ska implementeras. Sedan löper utvecklingsarbetet och dagliga möten hålls inom utvecklingsteamet. Vid iterationens slut håller utvecklingsteamet en demo för produktägaren och intressenter/verksamhetsrepresentanter. Slutligen hålls ett återblicksmöte där man utvärderar och försöker ta lärdom av eventuella problem under den gångna iterationen.

Vid iterationens början (sprintplanering) förs arbetspunkter över från product backlog till sprint backlog. Vid denna tidpunkt måste backlogen vara detaljerad nog att utvecklings-teamet förstår vad som ska implementeras, hur det ska testas, och vad som krävs för att arbetet ska anses vara klart. Oftast sker en hel del justering, förklaring och tidsestimering under själva mötet. Produktägaren måste här vara beredd på att snabbt kunna omprioritera.

Under utvecklingsiterationen måste produktägaren vara tillgänglig att svara på frågor kring kravställningen, förtydliga arbetspunkter i backlogen osv. I övrigt arbetar produktägaren med att detaljera backlogen inför nästa sprint, dvs börjar om vid steg 2 med nästa process(er).

Under iterationens gång och i synnerhet precis vid slutet (sprintdemo och sprint-återblick) är det möjligt (och mycket troligt) att det uppkommer "lösningsekvenser", alltså nytt arbete som krävs i nästa iteration som en konsekvens/insikt från nuvarande iteration. Dessa förs in i och prioriteras mot hela backlogen.



Bilaga 1: Exempel – Kundtjänst

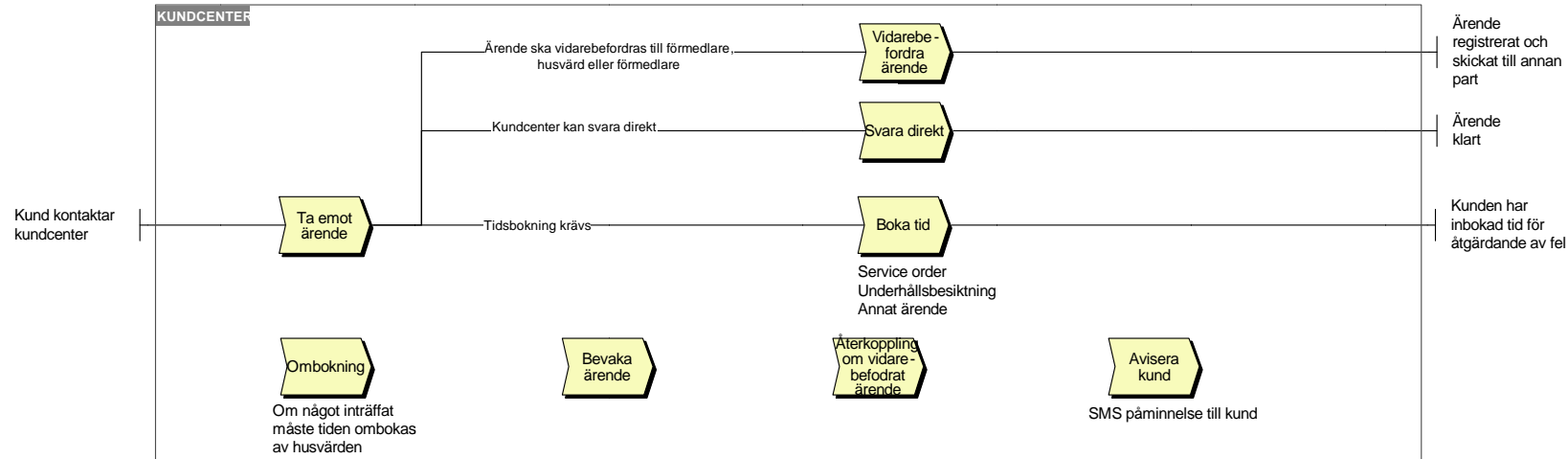
Nedan följer ett exempel på vad som tas fram i varje steg i en iteration och hur product backlog växer fram i respektive steg.

Steg 1. Kartläggning

Processkarta

version

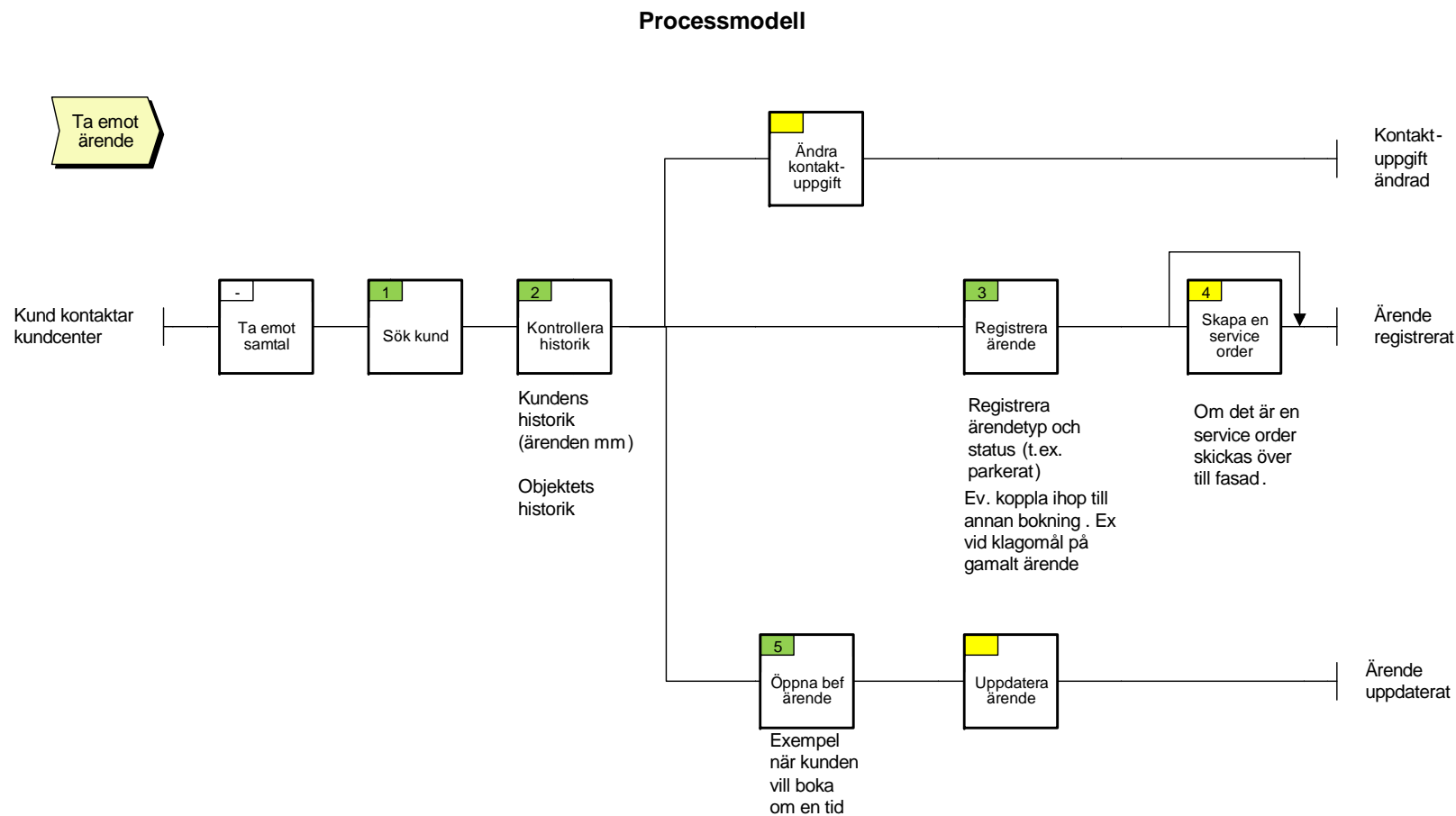
2010-12-15



Process	Resultat
Ta emot ärende	Ärende registrerat
Boka tid	Kunden har inbokad tid för åtgärdande av fel
Husvärd bokar om tid	Kunden har ny tid för åtgärdande av fel
Svara direkt	Ärendet avslutat
Vidarebefordra ärende	Ärende registrerat och skickat till annan part
Bevaka ärende	Ärende under bevakning

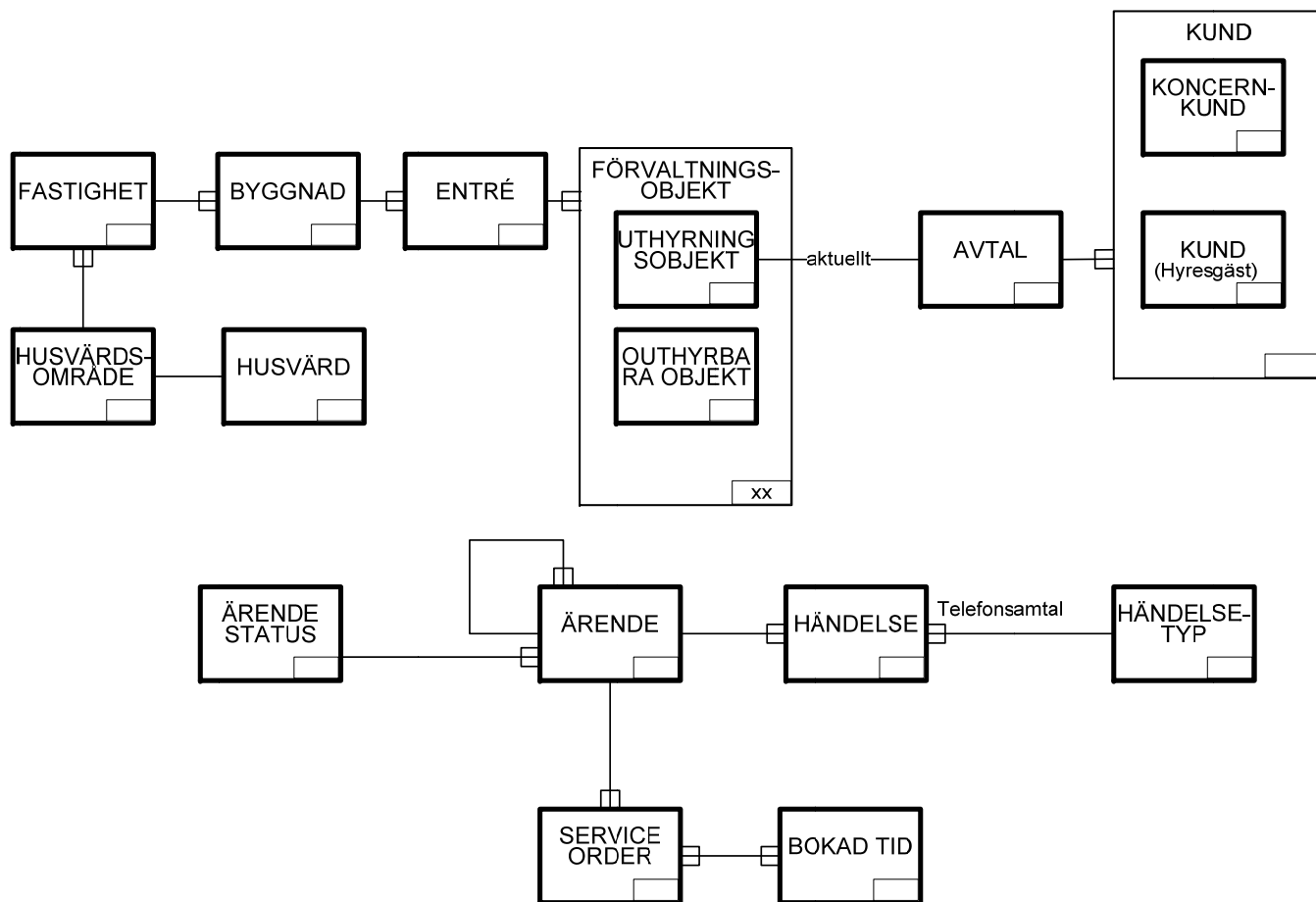
Steg 2. Processmodell och informationsmodell

Processmodell tas fram för den första (högst prioriterade) processen.



Obligatoriska aktiviteter markerades här som gröna, icke obligatoriska som gula.

Informationsmodell tas sedan fram för den information som krävs av aktiviteterna i processen. Informationsmodellen i sig genererar kanske inte användarberättelser som förs in i backlog, men den utgör ett viktigt underlag för tidsuppskattning och stöd för implementation i ett senare skede.



Typ	Process	Aktivitet	Akt#	Resultat
Aktivitet	Ta emot ärende	Sök kund	1	
Aktivitet	Ta emot ärende	Kontrollera historik	2	
Aktivitet	Ta emot ärende	Registrera ärende	3	Ärende registrerat
Aktivitet	Ta emot ärende	Öppna befintligt ärende	5	
Process	Ta emot ärende			Ärende registrerat
Process	Boka tid			Kunden har inbokad tid för åtgärdande av fel
Process	Husvärd bokar om tid			Kunden har ny tid för åtgärdande av fel
Process	Svara direkt			Ärendet avslutat
Process	Vidarebefordra ärende			Ärende registrerat och skickat till annan part
Process	Bevaka ärende			Ärende under bevakning

Typ	Process	Aktivitet	Akt#	Resultat	Som/om	vill jag	så att
Aktivitet	Ta emot ärende	Sök kund	1		Kundtjänst	Mata in namn/personnr/kundnr och få lista på matchningar	jag hittar rätt kund
Aktivitet	Ta emot ärende	Kontrollera historik	2		Kundtjänst	Visa kundens historik	få överblick
Aktivitet	Ta emot ärende	Registrera ärende	3	Ärende registrerat	Kundtjänst	Mata in ett ärende enligt kundens specifikation	serviceorder kan skapas utifrån informationen
Aktivitet	Ta emot ärende	Öppna befintligt ärende	5		Kundtjänst	Öppna befintligt ärende	ärendet kan kompletteras med ny information
Process	Ta emot ärende			Ärende registrerat			
Process	Boka tid			Kunden har inbokad tid för åtgärdande av fel			
Process	Husvärd bokar om tid			Kunden har ny tid för åtgärdande av fel			
Process	Svara direkt			Ärendet avslutat			
Process	Vidarebefordra ärende			Ärende registrerat och skickat till annan part			
Process	Bevaka ärende			Ärende under bevakning			

Icke-obligatoriska aktiviteter förs in, skrivs som användarberättelser och prioriteras mot hela backlogen:

Typ	Process	Aktivitet	Akt#	Resultat	Som/om	vill jag	så att
Aktivitet	Ta emot ärende	Sök kund	1		Kundtjänst initialt	Mata in namn/personnr/kundnr och få lista på matchningar	jag hittar rätt kund
Aktivitet	Ta emot ärende	Kontrollera historik	2		Kundtjänst	Visa kundens historik	få överblick
Aktivitet	Ta emot ärende	Registrera ärende	3	Ärende registrerat	Kundtjänst	Mata in ett ärende enligt kundens specifikation	serviceorder kan skapas utifrån informationen
Aktivitet	Ta emot ärende	Öppna befintligt ärende	5		Kundtjänst	Öppna befintligt ärende	ärendet kan kompletteras med ny information
Process	Ta emot ärende			Ärende registrerat			
Process	Boka tid			Kunden har inbokad tid för åtgärdande av fel			
Aktivitet	Ta emot ärende	Skapa en serviceorder			Kundtjänst	Skapa en serviceorder utifrån ett ärende	husvärd kan åtgärda fel
Process	Husvärd bokar om tid			Kunden har ny tid för åtgärdande av fel			
Aktivitet	Ta emot ärende	Ändra kontaktuppgift			Kundtjänst	Ändra kundens kontaktuppgifter i kundbilden i samband med registrering av ärende	kontaktuppgifter kan hållas aktuella på smidigt sätt
Process	Svara direkt			Ärendet avslutat			

Aktivitet	Ta emot ärende	Skriv avbokad i kalender		Kundtjänst	Markera bokning som avbokad i husvärds kalender, med "länk" från kundens historik / ärendebild	avbokning kan ske smidigt vid telefonkontakt med kunden
Process	Vidarebefordra ärende			Ärende registrerat och skickat till annan part		
Aktivitet	Ta emot ärende	Kontrollera historik	2		Visa historik för ett objekt, "länkat" från kundbild (objekt som är kopplade till kunden)	få överblick och kunna ge kunden information om objektet
Process	Bevaka ärende			Ärende under bevakning		

Steg 3. Tekniska bidrag

Typ	Process	Aktivitet	Akt#	Resultat	Som/om	vill jag	så att	Tid est.
Teknik						Sätta upp projekt och versionshantering		1
Teknik						Sätta upp byggserver och testmiljö		2
Teknik	<i>Ta emot ärende</i>					Integration (databaskoppling) för kundregister		3
Aktivitet	Ta emot ärende	Sök kund	1		Kundtjänst initialt	Mata in namn/personnr/kundnr och få lista på matchningar	jag hittar rätt kund	2
Teknik	<i>Ta emot ärende</i>				<i>Kundtjänst</i>	<i>Få kundbild efter sökning</i>	<i>information blir tillgänglig</i>	5
Aktivitet	Ta emot ärende	Kontrollera historik	2		Kundtjänst	Visa kundens historik	få överblick	3
Teknik	<i>Ta emot ärende</i>					<i>Databasmodell: Ärende</i>		2
Aktivitet	Ta emot ärende	Registrera ärende	3	Ärende registrerat	Kundtjänst	Mata in ett ärende enligt kundens specifikation	serviceorder kan skapas utifrån informationen	3
Aktivitet	Ta emot ärende	Öppna befintligt ärende	5		Kundtjänst	Öppna befintligt ärende	ärendet kan kompletteras med ny information	2
Process	Ta emot ärende			Ärende registrerat				
Process	Boka tid			Kunden har inbokad tid för åtgärdande av fel				
Aktivitet	Ta emot ärende	Skapa en serviceorder			Kundtjänst	Skapa en serviceorder utifrån ett ärende	husvärd kan åtgärda fel	3

Process	Husvärd bokar om tid		Kunden har ny tid för åtgärdande av fel			
Aktivitet	Ta emot ärende	Ändra kontaktuppgift	Kundtjänst	Ändra kundens kontaktuppgifter i kundbilden i samband med registrering av ärende	kontaktuppgifter kan hållas aktuella på smidigt sätt	2
Process	Svara direkt		Ärendet avslutat			
Aktivitet	Ta emot ärende	Skriv avbokad i kalender	Kundtjänst	Markera bokning som avbokad i husvärdens kalender, med "länk" från kundens historik/ärendebild	avbokning kan ske smidigt vid telefonkontakt med kunden	3
Process	Vidarebefordra ärende		Ärende registrerat och skickat till annan part			
Aktivitet	Ta emot ärende	Kontrollera historik	2	Visa historik för ett objekt, "länkat" från kundbild (objekt som är kopplade till kunden)	få överblick och kunna ge kunden information om objektet	2
Process	Bevaka ärende		Ärende under bevakning			