



UPPSALA
UNIVERSITET

STS12 010

Examensarbete 30 hp
April 2012

Appar eller ej inom Fleet Management-området

Organisatoriska och tekniska utmaningar
vid apputveckling

Taavi Rahnel
Boris Ristov



UPPSALA
UNIVERSITET

Teknisk- naturvetenskaplig fakultet
UTH-enheten

Besöksadress:
Ångströmlaboratoriet
Lägerhyddsvägen 1
Hus 4, Plan 0

Postadress:
Box 536
751 21 Uppsala

Telefon:
018 – 471 30 03

Telefax:
018 – 471 30 00

Hemsida:
<http://www.teknat.uu.se/student>

Abstract

Appar eller ej inom Fleet Management-området

Apps or no apps within the Fleet Management area

Taavi Rahnel

A recent trend in the technological evolution has been to increase the mobility of the end-user. A contributing factor is the fast rate of improvement in the smartphone area. This, in turn, has changed the expectations end-users have on the availability of services, which has allowed mobile applications, apps, to find their place in everyday life.

This study has been conducted at Scania's Fleet Management department that offers a web-based service where haulage contractors can administer, monitor and follow up on their vehicle fleets. The purpose of this study was to investigate if Scania's existing web service should be extended into the mobile area, and if so, what organizational and technical challenges app development imposes. To address this problem, a qualitative investigation was carried out with three main groups of respondents: comparable companies that have faced similar challenges, app development companies and the current users of the Fleet Management service. Additionally, a prototype application was developed to investigate the potential technical challenges that may occur. Combined with a theoretical framework the conducted research has led to recommendations on how to best take the step into the field of mobile devices.

This study shows that it is advisable to adopt the mobile technological trend since the positive effects of an app outweigh the potential costs. The technical changes within the field of smartphones occur at a rapid rate, which requires special technical skills. The development also requires iterative development methods where the end-user is included throughout the whole process.

Handledare: Anders Lundsgård
Ämnesgranskare: Roland Bol
Examinator: Elísabet Andrésdóttir
ISSN: 1650-8319, UPTEC STS12 010
Sponsor: Scania CV AB

Sammanfattning

En trend i den teknologiska utvecklingen på senare tid har varit att skapa en ökad rörlighet hos användaren. En bidragande faktor till detta är den snabba förbättringstakten hos de smartphones som i större utsträckning har funktionalitet som tidigare endast datorer hade. Detta har i sin tur förändrat de förväntningar slutanvändarna har på tjänsters tillgänglighet, vilket har lett till att produktiva mobila applikationer, appar, har funnit sin plats i vardagen.

Denna studie är genomförd på Scantias Fleet Management-avdelning som erbjuder en webbaserad tjänst där åkerier kan administrera, övervaka och följa upp sina fordonsflottor. Syftet med studien var att undersöka om en förlängning av Scantias existerande webbtjänst till den mobila sfären är rätt steg att ta och vilka organisatoriska och tekniska utmaningar en ingång inom apputveckling ger upphov till. För att bemöta denna problematik har en omfattande kvalitativ undersökning genomförts där tre huvudgrupper av respondenter har intervjuats: jämförbara företag som har ställts inför liknande utmaningar, apputvecklingsföretag som har förståelse för appområdet samt Fleet Management-tjänstens existerande användare. Dessutom har en undersökning av potentiella tekniska utmaningar vid apputveckling gjorts genom framtagning av en prototyp i form av en app. Den genomförda undersökningen har, i kombination med studiens teoretiska referensram, lett till rekommendationer om hur man bör bemöta steget till de mobila enheterna.

Denna studie visar på att det är lämpligt att anamma denna teknologiska trend då de positiva effekterna av en app väger upp kostnaderna. Apputveckling kräver ett tillvägagångssätt som skiljer sig från klassisk mjukvaruutveckling då slutanvändarens roll är mycket viktig under hela utvecklingsprocessen samt att en teknisk specialistkompetens behövs. En starkt iterativ utvecklingsprocess förutsätts och en proaktiv utvecklingsfilosofi krävs då tekniska förändringar inom smartphone-världen sker med stormsteg.

Förord

Denna uppsats kunde ej ha skrivits utan det stöd som har erhållits från flera håll.

Vi vill inleda med att tacka Anders Lundsgård, vår handledare på Scania som har varit till stor hjälp. Hans stöd och uppmuntran under resans lopp samt all den feedback vi har fått har hjälpt oss otroligt mycket.

Dessutom vill vi rikta ett stort tack till Peter Larsson på REIT-avdelningen som har hjälpt oss komma i kontakt med Scantias driftprovskunder runtom i Stockholmsregionen. Utan denna hjälp hade uppsatsens omfång varit märkbart mindre. Tack till de övriga medarbetarna på avdelningen REI som ställt upp och svarat på frågor och deltagit i diskussioner.

Ett stort tack riktas även till dem som ställde upp för intervjuer utanför Scania. Informationen som inhämtats från er har bidragit med en oersättlig del i studien.

Sist men inte minst vill vi även tacka våra handledare på respektive institutioner, Stefan Tongur på Industriell ekonomi och organisation på KTH samt Roland Bol på Institutionen för informationsteknologi på Uppsala Universitet. Utan era råd och konstruktiva kritik hade arbetet förmodligen ej fortlöpt på samma sätt.

Södertälje, januari 2012

Taavi Rahnel & Boris Ristov

Innehållsförteckning

1	Inledning	1
1.1	Scania Fleet Management som fall	2
1.2	Syfte	3
1.3	Avgränsning	4
1.4	Arbetsfördelning	4
1.5	Disposition	5
2	Metod	6
2.1	Arbetsflöde	6
2.2	Studie av Fleet Management-systemet	6
2.3	Informationsinsamling	7
2.3.1	Benchmarking	7
2.3.2	Apputvecklingsföretagens erfarenheter	9
2.3.3	Åkeriers åsikter	9
2.4	Prototyputveckling	10
2.5	Kunskapsansats	10
2.5.1	Intervjumetodik	11
2.5.2	Observationer	12
2.6	Litteraturstudie	12
2.7	Reliabilitet och validitet	13
2.8	Källkritik	14
3	Teoretiskt ramverk	15
3.1	Projektmodeller för mjukvaruutveckling	15
3.1.1	Agil mjukvaruutveckling som koncept	15
3.1.2	Rational Unified Process	16
3.1.3	Scrum	21
3.2	Lean-principer i mjukvaruutveckling	24
3.3	Outsourcing av mjukvaruutveckling	26
3.4	Tilläggstjänster för högkostnadsprodukter	28

3.5	Användarvänlighet.....	29
3.5.1	Mobila applikationer och användarvänlighet.....	30
4	Bakgrund om Scania.....	31
4.1	Scania-verksamheten	31
4.2	Scania Fleet Management	31
4.2.1	Fleet Management System	32
4.3	Scania Fleet Managements kunder	36
4.4	Scanias utvecklingsmodeller.....	37
4.4.1	Scania R&D Factory	37
4.4.2	Product Development Process	38
4.4.3	Software Development Process	39
4.5	Scania Fleet Managements konkurrenser.....	42
5	Apputveckling.....	44
5.1	Intervjuer med jämförbara företag.....	44
5.1.1	Förankrings- och förvaltningsprocessen.....	45
5.1.2	Utvecklingsmodell för apputveckling.....	47
5.1.3	Intern respektive extern utveckling.....	50
5.1.4	Användarvänlighet och slutanvändarens roll.....	52
5.2	Utveckling till mobila plattformar	53
5.2.1	Utveckling till iOS	54
5.2.2	Utveckling till Android.....	55
5.2.3	Multiplattformslösningar	56
5.2.4	Webbtjänster	57
5.2.5	Publicering av appar.....	57
5.3	Val av plattform	59
5.3.1	Mobila plattformar	59
5.3.2	Val av plattformsversion.....	59
6	Fleet Management-prototypen	61
6.1	Användandet av FMP-portalen	61

6.2	FM-appens potentiella funktionalitet	62
6.3	Utveckling av en Fleet Management-prototyp.....	62
6.4	Återkoppling från användarna	68
7	Analys	69
7.1	Att utveckla mobila appar	70
7.2	Intern eller extern utveckling av mobila appar	71
7.3	Utvecklingsprocessen.....	73
7.4	Tekniska val	75
7.5	Förvaltning och underhåll av app efter publicering	77
8	Rekommendationer	79
9	Slutord.....	82
9.1	Framtida utveckling	83
10	Litteraturförteckning	84
	Bilaga – Intervjumallar	I
	A - Mall för apputvecklingföretag	I
	B - Mall för driftprovskunder	III
	C - Mall för jämförbara företag.....	V

Figurförteckning

Figur 1: Illustration över studiens disposition	5
Figur 2: Illustration över arbetsflödet	6
Figur 3: Benchmarkingmodell (bygger på Karlöf, 2009 och Andersen & Pettersen, 1995)	8
Figur 4: Klassisk vattenfallsmodell inom mjukvaruutveckling, där risken ökar ju längre fram ändringar sker (bygger på Gibbs, 2006)	17
Figur 5: En bild på RUP-ramverket samt dess arbetsflöden och faser (Gibbs, 2006)	19
Figur 6: Scrum-modellen och dess tre övergripande faser (Abrahamsson et al., 2002)	22
Figur 7: Outsourcing-matris som hjälper beslutsfattare outsourca rätt processer (bygger på Quinn, 1999)	27
Figur 8: Organisationsschema över Scania och Fleet Management	32
Figur 9: Illustration över hur Fleet Management-datatrafik ser ut	33
Figur 10: Illustration över en trafikljusrapport i FMS	34
Figur 11: Illustration över kartfunktionen i FMS	35
Figur 12: Illustration över historiska data i FMS	35
Figur 13: Översikt över fördelningen av fordon bland åkerier i Europa (Eriksson, 2011)	36
Figur 14: Översikt över typiska kostnadsposter för ett åkeri i Europa (Eriksson, 2011)	37
Figur 15: Scania R&D Factory som illustrerar Scantias grundläggande kärnvärden (R&D Factory, 2010)	38
Figur 16: Delprocesserna i Scantias PD-process (Product Development Process, 2011)	39
Figur 17: Prognos för antalet sålda smartphone-enhet per plattform (bygger på Petty & Holley, 2011b). Notera att år 2013 och 2014 är egna uppskattningar	54
Figur 18: Illustration över fördelningen bland de nuvarande Android-versionerna (Android Developers - Platform Versions, 2012)	60
Figur 19: iPhone-storyboard för Fleet Management-prototypen	63
Figur 20: Loginvyn för Fleet Management-prototypen	64
Figur 21: Jämförelse av paketstorlek då REST respektive SOAP används	65
Figur 22: Kartvyn för Fleet Management-prototypen	66
Figur 23: Listvyn för Fleet Management-prototypen	67
Figur 24: Detaljvyn för Fleet Management-prototypen	67
Figur 25: Illustration över den analysmodell som har använts för studien	69

Tabellförteckning

Tabell 1: Sammanfattning av roller inom Scania Fleet Management	41
Tabell 2: Sammanfattning av intervjuobjekt samt företag	53
Tabell 3: Sammanfattning av intervjuobjekt samt företag	58
Tabell 4: Sammanfattning av intervjuobjekt samt företag	60
Tabell 5: Sammanfattning av intervjuobjekt samt driftprovskunder	68

Begreppsdefinition

Agil mjukvaruutveckling – Ett koncept som förespråkar enkelhet och snabbhet inom mjukvaruutveckling. Bland annat förespråkas återkommande kontakt med slutkunden.

Android – Operativsystem som används i flertal mobila enheter idag.

API – Application Programming Interface, en uppsättning regler för hur programvara kan kommunicera med annan programvara via funktionsanrop.

App – Kort för mobil applikation, en form av mjukvara som går att installera på avancerade mobiltelefoner, så kallade “smartphones”.

Artefakt – Inom RUP ett namn för de resultat som en utvecklingsroll använder sig av eller producerar i ett mjukvaruutvecklingssammanhang.

Backlog – Inom Scrum en lista som presenterar olika krav och nödvändiga funktioner som rangordnas efter betydelsen och den insats som krävs. Denna lista revideras kontinuerligt under projektets gång.

Fleet Management – Ledning av en organisations fordonsflottor. Fleet Management-området kan bland annat omfatta överblick av fordonsunderhåll, fordonstelematik (diagnostik och spårning) och ledning av förarna i flottan.

FMS – Fleet Management System, den Scania-standard som har framtagits för att handskas med fordonsspecifika data.

FMP – Fleet Management Portal, det webbaserade gränssnitt som Scantias kunder använder för att hantera sina fordonsflottor.

IDE – Integrated Development Environment, en mjukvarumiljö som möjliggör utveckling av mjukvara och som tillhandahåller alla nödvändiga funktioner.

iOS – Det operativsystem som används i iPhone, iPad och iPod.

Lean mjukvaruutveckling – En mjukvaruutvecklingsprincip som lånar från industrins lean-teorier och bygger på att eliminera slöseri och annat som har inverkan på produktkvaliteten.

Multiplattformlösningar – Mjukvara som är utformad för att kunna köras på så många olika plattformar som möjligt och som bygger på en enda gemensam kodbas.

Native – Används i samband som native-utveckling, native-app eller liknande. Syftar på utveckling eller app som är utvecklad i specifik miljö för just en viss plattform.

Proof-of-concept – En grov prototyp som används för att demonstrera ett framtaget koncept, i studien en tidig version av en app med begränsad funktionalitet.

REST – Ett protokoll för att hantera kommunikation mellan en klient och ett API. Datan hanteras av JSON, ett standardiserat format för att överföra data som läsbar text.

RUP – Rational Unified Process, ett projektramverk som inledningsvis skapades av Rational Software för att möjliggöra ledning och styrning av större mjukvaruprojekt. Företaget förvärvades av IBM år 2003.

Scrum – En lättviktig och agil mjukvaruutvecklingsmodell som behandlar själva utvecklingen som en svart låda, där utvecklarna behöver beakta förändringar som uppkommer i den rådande miljön.

Smartphone – En avancerad mobiltelefon som innehåller funktioner motsvarande handdatorer.

SOAP – Ett protokoll för att hantera kommunikation mellan en klient och ett API. Datat finns i XML-form.

Telematik – Ett område som integrerar telekommunikation och informatik. I fordonssammanhang handlar det om uppföljning av fordon.

1 Inledning

Detta kapitel berör bakgrunden till rapporten samt syftet och frågeställningarna. Dessutom presenteras avgränsningarna samt dispositionen för resten av rapporten.

Den teknologiska utvecklingen som idag sker har påverkat människans beteenden och beroenden. Datorn har gått från att vara stora enheter i datorhallar till att finnas i varje mobiltelefon. Det har skapat nya möjligheter och en större rörlighet. Då mycket i dag finns tillgängligt på Internet skapas krav från användarna, bland annat att de mobila enheterna som tillverkas i dag har anslutningsmöjligheter och att de tjänster som tidigare har varit tillgängliga på stationära datorer nu även skall vara tillgängliga på de mobila enheterna.

För organisationer som vill ligga i teknikens framkant gäller det att anamma nya och växande trender. Den explosionsartade utbredningen av smartphones har lett till att en ny marknad och nya målgrupper har öppnats upp för företag. Det amerikanska konsultbolaget Gartner förutspår att antalet sålda iOS- och Android-telefoner kommer att öka med 59 % mellan 2011 och 2012 och utvecklingen ser likartad ut för åren därefter (Petty & Stevens, 2011a). Företag ställer sig frågor angående vilka tjänster som är möjliga och lämpliga på de nya mobila plattformarna samt vilka förutsättningar som krävs för att verksamheten skall vara ledande på en föränderlig marknad. Detta har gjort att även företag som Scania, vars egentliga huvudsysselsättning är att tillverka lastbilar, nu intresserar sig för den mobila marknaden.

Ett av de stora bidragen dessa avancerade telefoner har kommit med är populariseringen av mobila applikationer, så kallade appar. Dagens smartphones har till stor del präglats av de handdatorer, Personal Digital Assistants (PDAs), som var populära under 90-talet. Dessa enheter erbjöd funktionalitet som fram tills dess endast återfanns i vanliga datorer, exempelvis hantering av kontaktinformation, planeringsverktyg samt en begränsad möjlighet att installera ytterligare mjukvara vid behov. Dock saknade de generellt telefonfunktionalitet (Jokela, 1999). Dessa enheters tidiga framgångar var ett bevis på att användarna eftersökte produktivitetsfrämjande verktyg i mobilt format. När PDA:n därefter kombinerades med funktionaliteterna från en mobil telefon, exemplifierat av Nokias framgångsrika Communicator-serie, skapades den nya smartphone-kategorin som medförde att efterfrågan på klassiska, telefonlösa PDA:n minskade kraftigt (Krazit, 2004). Den flexibilitet denna kategori av avancerade telefoner erbjöd skulle visa sig vara ett bestående fenomen, men det stora genombrottet kom när Apple slog upp portarna till App Store år 2008: inom loppet av ett år hade en och en halv miljarder appar laddats hem av iPhone-användare och ett nytt marknadssegment hade fötts (Kerris & Bowcock, 2009).

Trots att appar har existerat under en längre tid är det först på senare år som arbetet kring mer verksamhetsnära appar har tagit vid, detta då tillgängligheten av smartphones har ökat markant. Olika aktörer har börjat se värdet i att förlänga sina existerande tjänster till den mobila sfären, bland annat genom att skapa en lättanvänd mobil app av en existerande webbsida eller tjänst. Det har även medfört att olika företag har riktat in sig på att erbjuda specialistkompetens inom apputveckling för de företag som ej besitter kompetensen själva.

Då betydelsen av mobila appar, och mobila lösningar som helhet, ökar i takt med expansionen av smartphones finns ett behov i att undersöka hur teknikmedvetna företag bör agera. Mjukvaruutveckling i allmänhet är ett väl utforskat område, och mycket arbete har lagts på att ta fram generella och användbara principer för att förbättra utvecklingsprocesserna. Utvecklare har bland annat använt principer från tillverkningsindustrin för att skapa riktlinjer över hur mjukvaruutveckling kan ske på ett effektivare sätt (Poppendieck, 2006). Dessutom existerar väl etablerade och mer eller mindre agila projektmodeller som fokuserar just på hur mjukvaruutveckling bör genomföras, till exempel Rational Unified Process (RUP) (Gibbs, 2006) och Scrum (Schwaber & Beedle, 2008), modeller som kännetecknas av starka iterativa inslag. Mobila appar, och mobil utveckling i helhet, har dock fått lite uppmärksamhet i litteraturen: många studier fokuserar främst på de strikt tekniska aspekterna av utveckling för mobila plattformar eller på att genomföra marknadsundersökningar för de olika plattformarna. Således existerar en lucka i litteraturen som undersöker de organisatoriska aspekterna av en potentiell ingång i den mobila sfären och de unika tekniska krav som detta medför.

1.1 Scania Fleet Management som fall

Denna studie genomförs på uppdrag av Scania CV AB i Södertälje. Scania CV AB, härnäst Scania, är en tillverkare av stora transportfordon som lastbilar och bussar samt dieselmotorer för främst industriella och marina tillämpningar. Som ett led i den tekniska tjänsteutvecklingen har Scania infört en webbaserad plattform vid namn Fleet Management Portal (FMP) som används av åkerier för att förenkla övervaknings- och administrationsprocesserna av fordonsflottor. FMP är endast en del av ett komplett transportledningssystem som administreras av avdelningen Scania Fleet Management (Scania - Fleet Management, 2009). Tjänsten levereras i flera olika nivåer, och användaren har möjlighet att välja nivå efter behov.

Det övergripande syftet med Fleet Management-tjänsten är att erbjuda en heltäckande lösning som effektiviserar flottan och underlättar administrationen av ett åkeris fordon och förare. Den webbaserade portalen är dragplåstret i denna tjänst, med vilken trafikledare och andra roller kan övervaka alla fordon i flottan på ett lättöverskådligt sätt. Ett stort antal diagnostiska variabler övervakas av tjänsten, bland annat fordonets skick, bränsleförbrukningen och fordonets nuvarande och historiska positionsdata. Insamling av data görs via en fordonsdator vid namn Communicator, som installeras i samtliga Scania-lastbilar (Scania - Fleet Management, 2009).

Denna tjänst standardiserades av Scania under februari 2011, vilket innebär att alla nuvarande och kommande Scania-fordon i Sverige kommer att ha stöd för Fleet Management-tjänsten (Lundsgård, 2011a). Tjänstens betydelse kommer därmed att växa i takt med kundbasen, vilket även leder till ett ökat behov av att ständigt anpassa sig efter nya utvecklingar. Scanias intresse i att förlänga denna existerande webbtjänst till den mobila sfären blir således ett intressant fall att studera. Att utveckla mobila tillämpningar och lösningar som erbjuder mervärde för kunden är ett logiskt steg i den raska teknologiska utvecklingen, och går även hand i hand med Scanias kärnvärden (Scania - Kärnvärden, 2009). Då Fleet Management-sektionens fokus ligger på mjukvaruutveckling blir det även intressant att studera de förutsättningar och utmaningar som existerar, om de bör utveckla appen internt eller om en annan lösning är att föredra. Scania Fleet Management kan därmed användas som ett starkt studieexempel för hur en organisation bör göra för att ta steget över till den mobila världen.

1.2 Syfte

Syftet med studien är att undersöka om appar är något som avdelningen Fleet Management på Scania skall satsa på. Om så skall studien redovisa vilka tekniska och organisatoriska utmaningar som finns vid apputveckling. Studien skall även undersöka de olika möjligheter som finns vid apputveckling och ge vägledning i vilka beslut som kan tas för de olika möjligheterna.

Med bakgrunden och syftet som utgångspunkt har följande frågeställningar ställts upp:

- Vilka förutsättningar gäller för att Scania ska kunna ta fram en Fleet Management-app?
- Vilka organisatoriska och tekniska utmaningar finns vid utvecklingsförfarandet av mobila appar innan och efter lansering?
- Hur ska de utmaningar som finns beaktas för att en framgångsrik app ska kunna tas fram?

Svaret på och diskussionerna kring dessa frågeställningar skall mynna ut i och ge rekommendationer för avdelningen Fleet Management. Dessa skall ge en vägledning och lyfta fram områden som blir viktiga att ta ställning till vid en apputvecklingsprocess.

1.3 Avgränsning

Arbetet avgränsas till att undersöka de krav som ställs på en organisation som motsvarar Scania Fleet Managements nuvarande uppbyggnad, mer specifikt de krav som uppstår vid ett eventuellt steg från webb- till apputveckling. Berörda externa parter, såsom jämförbara företag, kommer användas vid benchmarking och intervjuer för att bland annat avgöra hur de har gått tillväga för att ta fram mobila appar. Dessutom kommer val av plattform och även vilken funktionalitet som är nödvändig i den mobila appen delvis bygga på de slutsatser som härleds ur kontakten med de potentiella användarna. Då examensarbetet har utförts på Fleet Management Portal-gruppen (REII) kommer de observationer som diskuteras främst att diskuteras utifrån REII:s perspektiv, då insynen i de andra grupperna är begränsad. Inga explicita kostnadskalkyler kommer utföras då diskussionen om extern eller intern utveckling uppkommer.

Då denna studie bygger på det fall som Scania Fleet Management erbjuder kommer ingen kvantitativ undersökning, via till exempel enkäter, genomföras. Det beror på att den existerande tjänsten i dagsläget huvudsakligen riktar sig mot åkeri- och fordonsägare och övriga roller som har ett intresse av att följa upp fordonsflottor. Urvalsgruppen blir därmed mycket mindre och detaljfrågor kan ställas direkt till de berörda rollerna. Dock har åkeriförarnas eventuella användning av tjänsten diskuterats, och resultaten kommer att redovisas i studien.

Den prototyp som tas fram kommer ej att utvecklas för Android, då Scania under sommaren 2011 har tagit fram en sådan app. Den alternativa plattformen som väljs skall alltså, ur ett tekniskt perspektiv, visa prov på att det går att skapa en verksamhetsnära mobil app som kan använda sig av den existerande tekniska grunden, i detta fall den existerande tekniska basen på Scantias Fleet Management-sektion.

Mjukvaruutvecklingsprojekt på Scania använder sig av RUP, vilket gör att studien fokuserar på att bedöma dess lämplighet vid utveckling av mobila appar. Projektmodeller för mjukvaruutveckling är ett omfattande ämnesområde, och studien avgränsas till att enbart presentera och diskutera de alternativa projektmodeller som används av de jämförbara företag som studerats. Då det har framkommit att Scrum är den utvecklingsmodell som har använts av intervjuobjekten avgränsas arbetet till att behandla enbart denna alternativa modell.

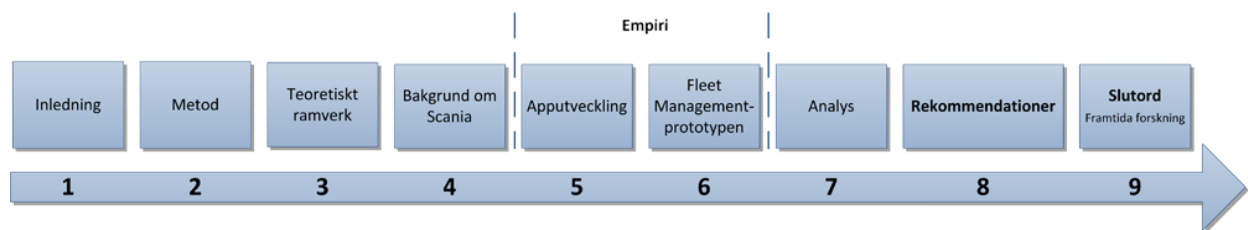
1.4 Arbetsfördelning

Detta arbete har genomförts av båda författarna på Scania Fleet Management-sektionen i Södertälje. Alla intervjuer som har genomförts som ett led i arbetet har gjorts av författarna tillsammans. Den iPhone-prototyp som har utvecklats, och som kommer diskuteras i detalj senare, har utförts av Rahnel. Den tekniska diskussionen i arbetet som bland annat berör prototypen och tekniska utmaningar i allmänhet har också gjorts av Rahnel, medan Ristov har stått för diskussionen kring de organisatoriska

utmaningarna som gäller vid apputveckling. Ristov har även haft en större inblandning vid presentationen av den teoretiska referensramen för arbetet. Valet av metoder och tillvägagångssätt samt den slutgiltiga analysen har gjorts av författarna tillsammans.

1.5 Disposition

Figur 1 illustrerar hur studien är disponerad. Kapitel 2 presenterar den metod som har använts för att genomföra studien och de olika intervjugrupper som har använts. I kapitel 3 finns det teoretiska ramverk som används för studien. Kapitel 4 presenterar det företag där studien har genomförts, Scania, samt Scania-specifika faktorer som har betydelse för det studerade fallet. I kapitel 5 har den empiriska intervjudatan samlats som har erhållits då studien genomfördes samt den empiri som berör utveckling till mobila plattformar. Kapitel 6 presenterar den insamlade informationen som knöt an till den apprototyp som togs fram under studiens gång. I kapitel 7 analyseras den empiriska datan med en återkoppling mot den uppställda teoretiska referensramen. Därefter följer de avslutande kapitlen där författarnas rekommendationer, slutord och förslag på vidare forskning summeras och presenteras.



Figur 1: Illustration över studiens disposition

2 Metod

Detta kapitel presenterar de vetenskapliga metoder som har använts för att genomföra studien. Avsnittet om arbetsflöde beskriver strukturen på arbetet och därefter redovisas arbetssättet för var och en av intervjugrupperna i separata avsnitt.

2.1 Arbetsflöde

För att nå bästa resultat på ett effektivt, systematiskt och övergripande sätt delades arbetet in i olika faser. Uppdelningen har varit möjlig då det har funnits tydliga delar som arbetet har bestått av. Ordningen av delarna har skett naturligt då en viss kunskap från en tidigare fas varit nyttig i ett senare skede. För att planera arbetets gång och få en bättre överblick över vad som skall göras när skapades i ett tidigt skede en tidsplan som beskriver vilka som är de viktigaste delarna i arbetet och när de skall utföras. De olika steg som arbetsflödet innehåller illustreras i figur 2. När alla de presenterade delarna har färdigställts kan dessa resultat analyseras och därefter kan slutsatser dras.



Figur 2: Illustration över arbetsflödet

2.2 Studie av Fleet Management-systemet

För att erhålla en grundläggande kunskap om processerna och systemen som sektionen för Fleet Management på Scania arbetar med var det första naturliga steget att studera de metoder och processer som idag används på sektionen. För att bekanta sig med de olika arbetssätten har intervjuer utförts med olika individer inom Fleet Management-grupperna. Anledningen till att dessa intervjuer genomfördes var att klargöra de generella frågor som fanns angående organisationen, de arbetsprocesser och -modeller som används samt bedöma det intresse som finns för apputveckling. Intervjuer har skett med personer som arbetar i olika grupper och som därmed har olika arbetsuppgifter. Detta för att få en mer mångsidig bild av vilka möjligheter och krav som finns internt för en apputveckling.

En annan viktig anledning var att klargöra hur den aktuella Fleet Management-tjänsten fungerar, Fleet Management System (FMS). Kunskap om hur det utvecklade systemet fungerar har även tagits in genom att använda de tjänster som FMS tillhandahåller sina kunder. Det har lett till en förståelse kring FMS egentliga funktion och dess nytta för kunden. Som ett led i detta arbete har den portal som finns tillgänglig för kunder använts och studerats.

2.3 Informationsinsamling

Efter studien av Fleet Management och FMS utfördes en informationsinsamling från olika parter för att få en insikt i området om mobila appar och dess möjligheter. Denna informationsinsamling går att dela in i tre separata delar där de olika delarna har haft som syfte att få kunskap inom olika områden.

2.3.1 Benchmarking

Benchmarkingen består av en studie av konkurrenters existerande mobila appar samt kontakt med jämförbara företag. Det finns ett flertal metoder som ger en vägledning i hur en process bäst skall startas upp och utföras. Valet av *benchmarking* gjordes på grund av att metoden kan appliceras för både strategiska och operativa frågor (Karlöf, 2009). Strategi definieras som en produktutvecklingsprocess med beslut där processen skapar ett resultat som bidrar till ett mervärde hos produkten. Den operativa delen handlar mer om de mer direkta frågorna som rör vid konkreta faktorer som är mätbara. Genom benchmarking kan referenspunkter därmed skapas som kan utgå ifrån (Karlöf, 2009). Med benchmarking menas i denna studie insamlandet av information hos externa parter för att se hur en process bäst skall genomföras på nytt och inte en jämförelse mellan den egna processen och externa parter liknande processer.

Oavsett vilken konkurrenskraft som finns inom branschen är det viktigt att sätta upp realistiska mål. Genom benchmarking går det att sätta upp relevanta mål för marknadskraven och metoden passar bra både för tillverknings- och tjänsteföretag (Bendell, Boulter & Kelly, 1993). Då det idag inte finns någon apputvecklingsprocess på Scania Fleet Management ger en benchmarking en vägledning till hur processen skall gå till och vilka målen skall vara om utvecklingsprocessen inleds.

Att utföra en extern benchmarking genom att studera konkurrenternas tillvägagångssätt i liknande situationer ger en inblick i de viktigaste faktorerna inom ämnet. Konkurrenterna är de företag som har en existerande Fleet Management-tjänst. Genom att observera konkurrenterna ges en inblick i de mest väsentliga faktorerna (Karlöf, 2009). Vid en utvidgad funktionell benchmarking som sträcker sig utanför konkurrenterna går det att hitta faktorer hos andra företag som organisationen själv kan implementera och som skall ge en försprång gentemot konkurrenterna. Dessa företag kan vara minst lika användbara som referenser vid en utvidgad benchmarking. Det hela slutar upp i att benchmarkingen handlar om att se hur liknande aktörer gör, lära sig av detta och göra likadant med tillägg från egen kreativitet eller utombranschorganisationer (Karlöf, 2009).

Observationer av konkurrenters appar

Genom att undersöka vad konkurrenterna gjort inom området, och se att de presterar väl ökar den egna ambitionsnivån vilket i sin tur leder till ökat lärande. Det i sig är en faktor till långsiktig framgång (Karlöf, 2009). De studerade apparna är FleetBoard av Daimler, WEBFLEET Mobile av TomTom,

MiX Telematics av MiX Mobile och Rapide av Masternaut. Apparna har studerats via deras funktioner, skärmdumpar och även testats i demolägen om sådana möjligheter har funnits.

Utveckling av appar hos jämförbara företag

Målet med de genomförda intervjuerna har varit att skapa sig en bild av hur jämförbara företag till Scania har gått tillväga då de valt att skapa lösningar till mobila produkter. Frågorna som ställdes till företagen finns bifogade som bilaga C. Med jämförbara företag till Fleet Management-sektionen menas verksamheter som har en existerande lösning på webben och som valt att komplettera den med en lösning till mobila enheter. Den här delen av studien har gjorts via intervjuer med personer ansvariga för eller delaktiga i utvecklingen av lösningarna till de mobila plattformarna. De företag som studerats i fråga är SEB, SJ, Svenska Spel, Sveriges Åkeriföretag, SVT, Taxi Stockholm och Scania där en app har tagits fram av en annan avdelning. Alla intervjuer har skett på plats hos respektive företag förutom SJ där frågorna istället besvarades via e-post och Scania där en telefonintervju genomfördes. Denna informationsinsamling har gjorts för att lära sig just hur de olika företagen har gått tillväga då de tagit beslutet att utveckla appar men även för att se vilka problem de stött på och hur de löst dessa.

Benchmarkingmodell

För benchmarkingen har en modell tagits fram som har fem olika faser och bygger på modeller av Karlöf (2009) och Andersen & Pettersen (1997). Modellen presenteras i figur 3.



Figur 3: Benchmarkingmodell (bygger på Karlöf, 2009 och Andersen & Pettersen, 1995)

I det första steget, planering, utformas en benchmarkingmetod där gruppen har en gemensam uppfattning i frågan. Här måste det även klargöras vad som egentligen skall benchmarkas. I nästa steg, sökande, gäller det att hitta jämförbara företag. Under informationsinsamlingen genomförs studier av konkurrenters appar och intervjuer utförs med de olika jämförbara företagen. Denna information som samlats in analyseras som därefter bidrar med lärdomar.

I slutändan skall benchmarken ge resultat. De erfarenheter som har inhämtats skall summeras och ge en effektivisering av verksamheten samt en kompetensutveckling. Gemensamt skall dessa faktorer båda öka den strategiska kompetensen och den operativa effektiviseringen. Hela den här processen kallar Karlöf (2009) för *benchlearning*. Konkret skall den information som samlas in ge Scania vägledning i hur de bör agera i frågan om appar skall utvecklas och i sådant fall hur utvecklingen bör ske.

2.3.2 Apputvecklingsföretagens erfarenheter

Då sektionen för Fleet Management inte har någon tidigare erfarenhet av apputveckling utfördes intervjuer med fyra olika apputvecklingsföretag. Detta gjordes för att få insikt i hur apputvecklingsprojekt i praktiken ser ut. De kontaktade företagen var Kondensator, Decuria, The Mobile Life och SWAD, som numera heter Springworks. Det finns ett stort urval av företag som utvecklar appar men då företag att kontakta söktes lades stor vikt på att de skulle ha utvecklat verksamhetsnära appar som har ett större syfte än endast marknadsföring. Det var även av vikt att företaget hade utvecklat appar med djupgående funktionalitet som bidrar med affärsnytta eller är allmänt verksamhetsnära. Företagen valdes då de har expertkunskaper inom apputveckling och då de kan bistå med erfarenheter av olika utvecklingsprocesser och de olika plattformarna samt även insikt i hur ett samarbete mellan en beställare och en utvecklare ser ut. Intervjuerna skedde på plats hos respektive företag med VD:n eller andra personer som har insikt i de olika uppdrag som utförts av företaget. Ämnen som diskuterades var av både tekniskt och organisatoriskt slag. Frågorna finns bifogade som bilaga A. Förutom att responsen från apputvecklarna hjälper för att få insikt i apputvecklingstekniker och processer ger den en insikt i vilka svårigheter som kan uppstå och som möjligtvis måste bemötas.

2.3.3 Åkeriers åsikter

De som den potentiellt utvecklade appen riktar sig mot är trafikledare på åkerier som är kunder till Scania. Därför var det av stor vikt att få in åsikter från dem som verkligen är de möjliga användarna av appen. Fleet Management-sektionen har i dag åkerier som är driftprovskunder och som agerar i en roll som testare av nyare produkter och lösningar. De kontaktade driftprovskunderna var Scania Transportlaboratorium, Scania-Bilar Sverige, CeEss Transport, Stjärnströms International och LTD Spedition. Personer som har varit delaktiga i trafikledningen har intervjuats för att få en insikt i hur de arbetar i stort och hur de använder sig av den existerande portalen idag och gav en inblick i hur appen skulle kunna underlätta deras dagliga arbete. Genom att samla in dessa data går det att få en insikt i vilka krav användarna har på en app och utforma appen efter dessa i fråga om funktioner och användargränssnitt. I studiens senare skede, då det fanns en prototyp att visa upp, gjordes detta för åkerierna som besöktes igen för att få en återkoppling. De frågor som användes för intervjuerna finns bifogade som bilaga B.

I denna studie intervjuades endast ansvariga för uppföljning av fordonsflottan, till exempel fordonsägarna eller trafikledarna. Eftersom tjänsten i dagsläget inte erbjuder någon specifik funktionalitet som riktar sig till förarna lades fokus på att tala med den existerande användargruppen.

2.4 Prototyputveckling

Prototypen för en app utvecklades i iOS till iPhone och iPad. Valet av iOS berodde på att de dominerande plattformarna idag är Android och iPhone och att en Androidprototyp sedan tidigare har utvecklats. Prototypens syfte var att visualisera de olika funktioner som idag är de mest använda av trafikledare samt även undersöka och se vilka tekniska krav utveckling av iOS-appar ställer på Fleet Management-sektionen.

En mindre prototyp utvecklades även i HTML5 med jQuery Mobile och PhoneGap. Det primära målet med denna prototyp var att hitta begränsningar som kommer av att utvecklingen sker i HTML5 på en gemensam kodbas för flera plattformar och se vilka krav som ställs vid utveckling med en gemensam kodbas. En mindre studie gjordes av de möjliga lösningarna där en kodbas utgör grunden för appar till flera plattformar och valet på jQuery med PhoneGap togs på grund av flera anledningar. jQuery Mobile bygger på HTML5 och har en mindre inlärningskurva än alternativen. Systemutvecklarna på Fleet Management-sektionen arbetar idag med webb och intryck från de olika intervjuerna med personer insatta i ämnet påverkade även valet.

Prototypen är utvecklad med en throw-away-prototypmetod. Med throw-away-prototyp menas en prototyp som i ett senare skede troligtvis inte kommer att lanseras i en produktionsmiljö. Efter att de primära kraven har samlats in skapas en prototyp som uppfyller kraven och som presenteras för användare för återkoppling. Prioriteringen i utvecklingen ligger på snabbhet och huvudsyftet är att testa koncept istället för att implementera dem på bästa sätt (Crinnion, 1991). Med snabb utveckling och täta interaktioner med användaren skall brister i systemet hittas i ett tidigt skede och därmed behöver inte ändringen bli stor, vilket kan jämföras med om bristen skulle hittas senare och då stora förändringar måste genomföras för att lösa problemet, vilket blir kostsamt. Tanken är därmed aldrig att en throw-away-prototyp skall lanseras och därmed skall det inte ställas för höga krav på den. I slutändan skall denna typ av prototyp hjälpa en utvecklare att ta fram de slutgiltiga kraven på systemet.

2.5 Kunskapsansats

Det finns två huvudsakliga undersökningsformer för insamling av data, *kvalitativa* och *kvantitativa* undersökningar. Kvalitativa undersökningar kännetecknas av att datainsamlingen sker på ett främst interpretivistiskt sätt, bland annat genom att använda metoder såsom djupintervjuer, fokusgrupper och observationer. Genom att använda sådana metoder erhålls rik och djupgående data som dock kan präglas av subjektivitet eller av den naturliga miljö studien har genomförts i. Dessa resultat kan däremot oftast enbart generaliseras till jämförbara miljöer och företag. Kvantitativa undersökningar, å andra sidan, strävar efter att producera exakta, kvantifierbara och objektiva resultat som därefter kan generaliseras från en stor sampelgrupp till en större population (Collis & Hussey, 2009).

Denna studie använder sig av kvalitativa tillvägagångssätt av flera skäl. Som tidigare nämnts är det största skälet att Scantias Fleet Management-tjänst som den ser ut idag ej riktar sig till förarna i fordonsflottorna. Sampelgruppen blir således mycket mindre och det blir därmed enklare att ta kontakt med de potentiella användarna, trafikledare och liknande roller, direkt för att erhålla mer nyanserad och detaljerad information. Då studien även ämnar att utröna processförfaranden och tillvägagångssätt i större detalj ger en kvalitativ undersökning den eftersökta informationen. Det är främst “hur?”- och “varför?”-frågorna som är av värde i denna studie, och med Scania Fleet Management som utgångspunkt ter sig valet av en kvalitativ undersökning naturlig.

2.5.1 Intervjumetodik

De intervjuer som gjordes för denna undersökning var semistrukturerade, vilket innebär att förberedda frågor ställdes till respondenten som därefter lämnades utrymme att besvara efter egna erfarenheter och åsikter. Genom att använda denna intervjuteknik erhöles djupgående kvalitativ data, särskilt om processer och arbetsmodeller. Följdfrågor för att klargöra de ämnen som diskuterades ställdes även vid behov (Collis & Hussey, 2009; Kvale & Brinkmann, 2009).

Frågorna planerades i förväg och skickades till respondenterna innan intervjutillfället så att en möjlighet till förberedelse fanns. De planerade frågorna utgjorde också manus till intervjuerna så att diskussionerna ej skulle avvika från det tematiska området (Kvale & Brinkmann, 2009). De flesta intervjuerna bokades in för en timme men varierade i längd. De intervjuer som genomfördes utanför Scania spelades in och transkriberades, och vid behov kontaktades respondenten återigen för att stämma av oklarheter. De intervjufrågor som skapades kontrollerades att de stämde med de rekommendationer som Eriksson och Wiedersheim-Paul (2006) presenterar. Exempelvis kan det handla om att inte ställa ledande frågor och att inte ställa flera frågor samtidigt. Se bilagor för kompletta listor över de genomförda intervjuerna. Urvalsstorleken fastställdes främst genom att använda konvergensprincipen som säger att intervjuer skall fortgå tills inget nytt kan inhämtas från fortsatta intervjuer (Sörqvist, 2000).

Beroende på intervjuat företag har de generella frågeställningarna kunnat utvidgas med mer specifika frågor som varit mer lämpliga och aktuella. Exempelvis har frågor kring säkerhet varit extra aktuella då möten har skett med SEB och Svenska Spel då dessa företag måste ha hög prioritet på säkerheten.

2.5.2 Observationer

En annan form av kvalitativ datainhämtning är att använda sig av *observationer*. Genom att använda sig av denna inhämtningsmetod inhämtas data då människor observeras i deras naturliga miljö, exempelvis de systemutvecklare som arbetar på Fleet Management-sektionen. Det finns huvudsakligen två typer av observationer, *icke-deltagande* och *deltagande observationer*. Med icke-deltagande observation menas att de som genomför observationen ej involveras i de aktiviteter som undersöks. Dock måste forskaren explicit fråga de observerade personerna om lov ifall undersökningen kräver att de observerade förfarandena dokumenteras. Med deltagande observation menas att de som genomför observationen fullt ut deltar i de undersökta aktiviteterna. Målet med båda typerna är att erhålla en detaljerad insikt i och förståelse för det som observeras (Collis & Hussey, 2009).

Eftersom författarna ägnade en stor del av arbetstiden på Fleet Management-sektionen i Södertälje kunde viss kvalitativ information inhämtas från observationer. Författarna deltog i de gruppmöten som gruppen hade veckovis och dessutom hölls möten med Scania-handledaren cirka två gånger per vecka. Planeringen på sektionen sker visuellt och författarna hade tillgång till de pulstavlor som används för all projektplanering samt övrig information som fanns tillgänglig i lokalen. Allt detta medförde att författarna fick en bättre möjlighet till att förstå hur Fleet Management-projekt genomförs, samt även bilda sig en uppfattning om den generella inställningen till mobila appar. Eftersom författarna till viss grad involverades i gruppens aktiviteter kan det argumenteras för att deltagande observation användes.

Kritik som framförs mot denna metod är bland annat att kontrollen över datainsamlingen är obefintlig. De observationer som erhålls kan färgas av observatörernas partiskhet, och observatörerna kan även helt missa att fånga in viktiga observationer på grund av distraktioner (Collis & Hussey, 2009). Av denna anledning är denna datainhämtningsmetod mer av ett komplement än något som författarna understödjer slutsatserna på.

2.6 Litteraturstudie

En omfattande litteraturstudie har skett under studiens lopp och grundar sig i de uppställda frågeställningarna och de avgränsningar som dessa har medfört. När en klar bild av den teoretiska referensramen erhöles användes lämpliga nyckelord för sökning av litteratur i databaserna på respektive lärosäte samt Google Scholar. Ett par exempel på de sökord som användes är "*agile software development*", "*rational unified process*", "*agile methods*", "*mobile application*", "*application development*" och "*outsourcing software development*".

En kort genomgång av tidigare studier och arbeten kring det valda ämnesområdet gjordes i publikationsdatabasen DiVA, detta för att skapa en förståelse över hur tidigare undersökningar har genomförts och de resultat som har framkommit från dessa. Denna genomgång visade på att utförda

undersökningar om mobila appar främst var av rent teknisk karaktär, exempelvis att framställa en app som uppfyller ett specifikt syfte åt en uppdragsgivare.

Då Scania har ett omfattande intranät vid namn Scania InLine har även den använts för att inhämta Scania-specifik dokumentation. Detta intranät har huvudsakligen använts för att erhålla detaljer kring de modeller och processer som används på de mjukvaruorienterade grupperna.

2.7 Reliabilitet och validitet

Då denna undersökning främst bygger på ett kvalitativt tillvägagångssätt behöver en diskussion kring studiens reliabilitet och validitet göras. *Reliabilitet* innebär att samma utfall uppstår då samma typ av mätning genomförs gång på gång, oavsett vem som upprepar testet. Därmed undersöks en mätningens tillförlitlighet: om samma procedur ger upphov till olika resultat är mätningens reliabilitet låg. *Validitet*, å andra sidan, är den grad av korrekthet som mätningen ger upphov till. Forskaren skall således mäta det som verkligen skall mätas (Kirk & Miller, 1986). Reliabiliteten hos kvantitativa mätningar påverkas av precisionen i undersökningen, men en kvalitativ undersöknings validitet påverkas av att forskaren kanske ser det han eller hon vill se (Kirk & Miller, 1986).

Eftersom kvalitativa intervjuer utgör kärnan i undersökningens metod kommer reliabiliteten att vara låg, då svar på “hur”- och “varför?”-frågor är svåra att replikera mellan olika individer. Validiteten kommer dock att vara hög då lämpligt utformade intervjufrågor kommer ställas till respondenterna. För att testa undersökningens reliabilitet och validitet kommer *triangulering* utföras. Triangulering innebär att flera insamlingstekniker används vid insamlingen av data för att stärka tillförlitligheten (Golafshani, 2003). Påståenden gjorda av respondenter kommer därmed verifieras mot information som erhålls från annat håll, exempelvis tidigare genomförda studier, observationer eller existerande teori. Då viss risk för partiskhet finns blir detta förfarande viktigt. En brist som vanligen uppstår är att endast ett företag analyseras, till exempel Scania, men då vi utgår från ett flertal jämförbara företag har ett försök till att minska osäkerheten gjorts. Eftersom mycket av den erhållna informationen bygger på bland annat teknisk data och välgrundade projektmodeller går det att stämma av denna information mot existerande undersökningar och teori.

2.8 Källkritik

En medvetenhet finns om risken för att de apputvecklingsföretag som intervjuats har ett egenintresse i att främja sin egen roll i den mobila utvecklingsmiljön, vilket kan leda till partiska svar. Den största subjektiviteten finns troligtvis i svaren som rör ifall appen skall utvecklas av en extern leverantör eller av Scania själv.

Det finns också en risk i att de driftprovskunder som har kontaktats har en syn på Fleet Management-tjänsten som ej överensstämmer med Scantias verkliga kunder. Driftprovskunderna har en skyldighet att rapportera fel som uppdagas under användningen av tjänsten, men får i övrigt tillgång till tjänsten utan extra kostnad.

Även de intervjuer som skett med jämförbara företag som utvecklat en app utifrån deras webbportal ses kritiskt på då de kanske inte vill visa sina svagheter och de misstag de begått.

Eftersom mobila appar och mobil utveckling i stort är ett nytt och relativt outforskat fenomen finns det en risk i att den litteratur som existerar ej är väl beprövad. Studien kommer därmed till stor del bygga på beprövad litteratur från angränsande ämnesområden som just klassisk mjukvaruutveckling.

3 Teoretiskt ramverk

Detta kapitel presenterar det teoretiska ramverk som har använts för att kunna genomföra analysen av det erhållna empiriska materialet. Projektmodeller och lean-principer inom mjukvaruutveckling, samt användarvänlighet är de övergripande områden som behandlas i kapitlet.

3.1 Projektmodeller för mjukvaruutveckling

Genom årens lopp har en uppsjö av olika modeller framtagits för hur mjukvaruutveckling bör gå till. Dessa modeller skiljer sig främst i hur rollfördelningen i projektteamen bör se ut, hur uppföljningen och planeringen bör gå till samt hur kraven inom projekten bör hanteras (Abrahamsson et al., 2002).

I nästföljande avsnitt diskuteras konceptet för agil mjukvaruutveckling och i de efterföljande avsnitten presenteras två utvecklingsmodeller, Rational Unified Process (RUP) samt Scrum, varav den förstnämnda används på Scania i dagsläget. Agil mjukvaruutveckling som koncept presenteras då det ligger som grund för flertalet utvecklingsmetodiker. Scrum-modellen presenteras då det har framkommit att intervjuobjekt i området använder denna modell eller en modifierad variant av den.

3.1.1 Agil mjukvaruutveckling som koncept

Vid användning av termen agil mjukvaruutveckling refereras den grupp av utvecklingsmetoder som lägger tonvikt vid enkelhet och snabbhet. Under 1990-talet accelererade konceptualiseringen av lättviktiga modeller vars främsta syfte var att motverka de problem tidigare utvecklingsmodeller hade. De äldre modellerna ansågs vara för administrativt tunga och byråkratiska, och följde vattenfallsmodellen för slaviskt. Utvecklare ville gå ifrån den stora fokusen på att utföra omfattande kravanalysen i början av ett projekt till att låta kraven växa fram mer dynamiskt under arbetets lopp. Således började iterativa modeller med en större tonvikt på feedback att användas mer och mer (Larman & Basili, 2003).

Den ökande användningen av lättviktiga, iterativa modeller ledde till att framstående mjukvaruutvecklare år 2001 samlades och publicerade riktlinjer för vad som skall klassas som agil mjukvaruutveckling. Dessa riktlinjer, "Manifest för Agil systemutveckling", har sammanfattats som följer:

- **"Individer och interaktioner** framför processer och verktyg"
- **"Fungerande programvara** framför omfattande dokumentation"
- **"Kundsamarbete** framför kontraktsförhandling"
- **"Anpassning till förändring** framför att följa en plan"

"Det vill säga, medan det finns värde i punkterna till höger, värdesätter vi punkterna till vänster mer." (Kniberg, 2001)

Den första punkten syftar till att det är viktigt att få ett väl fungerande och sammansatt team, en faktor som leder till bättre lösningar och slutresultat. Den andra punkten förespråkar att teamet iterativt och kontinuerligt strävar efter att släppa testad mjukvara för att förbättra arbetsflödet. Den tredje punkten belyser vikten av att ha återkommande kontakt med kunden för att minimera risken för dispyter samt för att kontinuerligt klargöra de rådande kraven. Den sista punkten menar att krav och tekniska förutsättningar kommer förändras under utvecklingsprojektets livscykel, och att det då är nödvändigt att ha de resurser som krävs för att fortlöpande kunna utvärdera situationen och genomföra förändringar vid behov (Abrahamsson et al., 2002).

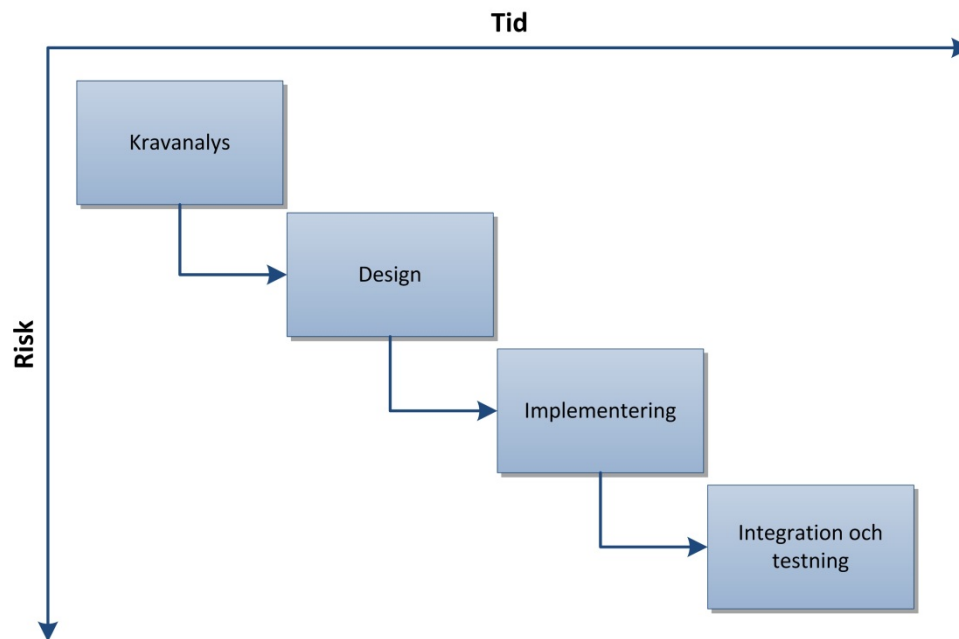
När mjukvaruutvecklingen anses vara *inkrementell* (små mjukvarusläpp med snabba cykler), *kooperativ* (kunden och utvecklarna samarbetar kontinuerligt), *rättfram* (metoderna är dokumenterade, lätta att lära sig och anpassningsbara) samt *adaptiv* (förändringar kan genomföras även i senare stadier) existerar en utvecklingsmetod som kan klassas som agil. Abrahamsson et al. (2002) betonar även att denna typ av utveckling är av värde inom volatila områden såsom just mobil apputveckling, där kundens krav kan förändras snabbt eller där initiala krav kanske ej existerar.

För att öka chansen till ett framgångsrikt projektresultat föreslår Cockburn (2007) bland annat att projektteamen bör bestå av mellan två till åtta medlemmar för att uppmuntra till ett kontinuerligt informationsutbyte. Alla former av feedback-cykler skall vara korta men återkommande och varje mjukvaruinkrement bör vara kort, exempelvis en månad, för att tillåta snabbare testning och bugfixande. Om möjligt bör teamet bestå av erfarna medlemmar, eftersom en erfaren utvecklare jobbar betydligt snabbare än oerfarna medlemmar. Dessa faktorer är därmed väsentliga för att teamet snabbt skall kunna producera resultat som därefter kan utvärderas av kunden eller av de tänkta användarna.

3.1.2 Rational Unified Process

Rational Unified Process (RUP) är ett omfattande utvecklingsramverk som används inom mjukvaruutvecklingsbranschen. Ramverket togs fram av Rational Software under 90-talet som ett svar på den fragmenterade utvecklingsmarknaden. Målet var att skapa en enhetlig metodologi för hur mjukvaruutveckling skall ske, vilket resulterade i att RUP växte fram till att bli en samling välgrundade best practice-principer. IBM förvärvade Rational Software år 2003 och RUP har genom deras framträdande roll förblivit en viktig modell i mjukvaruvärlden (Gibbs, 2006).

Många av de mjukvaruutvecklingsmodeller som växte fram var ett svar på de brister som uppfattades i den klassiska *vattenfallsmodellen*. I denna modell följs ett antal faser linjärt: inledningsvis klarläggs krav och behov analyseras, och därefter designas en lösning utifrån den genomförda förstudien. Projektplaneringen för hela projektet genomförs redan i början av projektet. Efter att en lösning har designats påbörjas implementeringsfasen, och projektet avslutas med att den implementerade lösningen testas innan den sätts i bruk. Varje fas i vattenfallsmodellen är fristående, och den föregående fasen väntas vara helt avslutad innan projektet flyter vidare till nästa fas (se figur 4).



Figur 4: Klassisk vattenfallsmodell inom mjukvaruutveckling, där risken ökar ju längre fram ändringar sker (bygger på Gibbs, 2006)

Modellens rigida struktur lämpar sig för projekt där kraven och även riskerna är tydliga redan i ett tidigt skede. Kraven på omfattande planering gör det även lätt att följa upp framstegen. Dock har stark kritik riktats mot denna modell. I mjukvaruprojekt är det ytterst sällan som det går att samla in och förstå alla krav och risker i ett tidigt skede, vilket gör att eventuella återgångar till tidigare faser blir kostsamma. Krav förändras under projektets gång, och även de tekniska förutsättningarna förändras vilket vattenfallsmodellen ej beaktar (Gibbs, 2006). Det är bland annat av denna anledning som den agila utvecklingsrörelsen tog vid.

RUP-modellen bygger på sex kärnprinciper vars mål är att minimera de risker som finns med mjukvaruutveckling. Dessa principer presenteras nedan (Kruchten, 2000):

- *Utveckla mjukvara iterativt*

Som ett svar på vattenfallsmodellens brister förespråkar RUP ett mer iterativt tillvägagångssätt. I varje iteration följs vattenfallsmodellen men i mindre skala. Således följs de faser som beskrevs tidigare, men bryter ner arbetet i mindre delar.

RUP-modellen menar att riskhanteringen förbättras genom att arbetet bryts ned i kortare iterationer: potentiella problem belyses i ett tidigt skede, särskilt då modellen manar till ett fortlöpande integrationsarbete istället för vid slutet av ett utvecklingsprojekt. Genom att iterera kan förändringar i exempelvis krav och teknologiska förutsättningar hanteras när de uppkommer. Korta, återkommande iterationer uppmuntrar även till förbättrad inläring då roller som vanligtvis utför sitt arbete i en senare fas involveras tidigt, till exempel testare. Då målet med varje iteration är att arbeta på en delmängd av

mjukvaran ökar även möjligheten till att producera moduler som kan återanvändas. Feedback från kravställare och slutanvändare kan också införlivas i arbetet i ett tidigt skede. RUP-förespråkare menar att kvaliteten på slutprodukten förbättras avsevärt om detta tillvägagångssätt används (Kruchten, 2000).

- *Hantera krav*

RUP-ramverket menar att det är viktigt att inse att kraven kommer att förändras. Det blir således ytterst viktigt att förtydliga de rådande kraven under projektets lopp. Nya krav tillkommer och gamla krav kan förändras. Kraven kan komma från intressenterna, till exempel slutanvändaren eller beställaren, och det blir då viktigt att förstå behoven. Krav kan även vara av teknisk art, till exempel att en specifik funktion skall finnas. Ett dokument som detaljerar visionen och bygger på de analyserade kraven skall tas fram och agera underlag för den fortsatta utvecklingen, och detta dokument skall granskas iterativt för att ta med de förändringar i kraven som kan uppkomma (Kruchten, 2000).

- *Använd modulbaserade lösningar*

Genom att bryta ner det slutgiltiga systemet till separata delar, moduler, kan varje del isoleras, utvecklas och testas var för sig. Dessa moduler kan utformas så att de blir återanvändbara, vilket leder till ökad produktkvalitet och lägre utvecklingskostnader. När modularisering används tydliggörs syftet med modulerna, och teamet får en möjlighet att reagera snabbare om modulen visar sig vara en felsatsning (Kruchten, 2000).

- *Modellera mjukvara visuellt*

Komplexa system bör enligt RUP brytas ner och förklaras i lättöverskådliga modeller. Dessa modeller används sedan för att snabbt få förståelse för hela systemet, och bidrar även till att hitta lösningar till problem på ett effektivare sätt. Det finns ett flertal olika modeller som ett mjukvaruutvecklingsprojekt bör beakta, exempelvis användningsfallsmodeller, affärsmodeller och testmodeller för enskilda komponenter (Kruchten, 2000).

- *Verifiera mjukvarukvaliteten kontinuerligt*

Enligt RUP är det viktigt att fortlöpande fastställa kvaliteten på arbetet som utförs, både på slutprodukten och på de ingående komponenterna. Alla avvikelser skall hittas och åtgärdas så tidigt som möjligt och det utförda arbetet skall ställas mot kraven kontinuerligt. Testning blir alltså en viktig och återkommande källa till feedback (Kruchten, 2000).

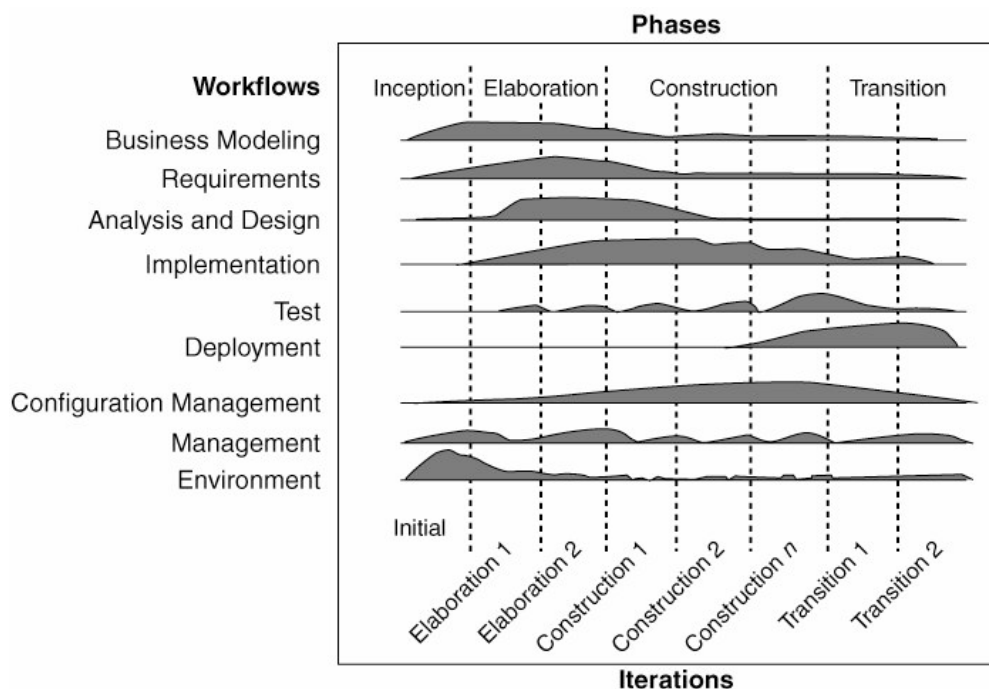
- *Kontrollera förändringar i mjukvaran*

I en iterativ och modulär utvecklingsmiljö kommer delkomponenter modifieras oberoende av varandra. Det blir därmed ytterst viktigt att säkerställa att de olika delarna i ett utvecklingsprojekt synkroniseras med varandra och att förändringar i mjukvaran kan följas upp, till exempel genom ett robust versionshanteringssystem. Det gäller även att kunna handskas med förändringar som uppkommer i kraven, i designen och i implementeringen (Kruchten, 2000).

Dessa principer utgör kärnan i ett RUP-projekt, som vanligtvis består av två dimensioner:

- Den *horisontella* dimensionen representerar tidsaxeln och visar hur de ingående delarna av projektet fortlöper. Denna dimension består av fyra sekventiella faser: *inception*, *elaboration*, *construction* och *transition*.
- Den *vertikala* dimensionen presenterar de arbetsflöden som finns i projektet och dessa grupperas logiskt, exempelvis testaktiviteter och implementeringsaktiviteter.

Se figur 5 för en illustration av ett typiskt RUP-projekt (Kruchten, 2000). Höjden på staplarna visar vikten av arbetsflödet under en viss fas.



Figur 5: En bild på RUP-ramverket samt dess arbetsflöden och faser (Gibbs, 2006)

Inception-fasen handlar om att specificera den gemensamma visionen för slutprodukten och att definiera projektets omfång, det vill säga bedöma vad som skall göras som en del av projektet. Bedömning av riskerna, kraven samt affärsnyttan är också viktiga steg i denna fas, och vanligtvis brukar prototyper användas för att demonstrera projektets giltighet. Denna fas är av större vikt då projektet ämnar bidra med något nytt, eftersom förarbetet då är mer omfattande.

Elaboration-fasen är den mest kritiska fasen av alla. Under denna fas skall en projektplan med tids- och resursuppskattningar utformas, alla funktioner som skall finnas med i slutprodukten skall definieras, alla krav från intressenterna skall behandlas samt mildra de risker som finns så långt det går. När denna fas är klar skall en stabil helhetsbild av projektet vara klar innan en övergång till utvecklingsfasen sker.

I Construction-fasen utvecklas alla funktionalitetsbringande moduler som sedan integreras i produkten. Modulerna skall uppfylla de ställda kraven och testas kontinuerligt under utvecklingen. De prototyper som användes inledningsvis skall med andra ord utvecklas till fullgoda produkter.

Transition-fasen fokuserar på leveransen av produkten till de ämnade slutanvändarna. Produkten skall börja användas i en produktionsmiljö så att alla fel och buggar kan åtgärdas. Funktionerna skall justeras baserat på den feedback som erhålls och användarna skall tränas upp vid behov. Merparten av uppföljningen sker i denna fas, och när fasen är avslutad har den slutgiltiga produkten levererats (Kruchten, 2000).

Varje fas avslutas med en milstolpe där projektets fortskridande utvärderas och om projektet behöver iterera inom fasen för att nå de ställda målen. Endast när en fas anses fullbordad fortsätter projektet till nästa fas.

Ur ett vertikalt perspektiv gäller det att definiera vem som gör vad och hur det skall göras. RUP använder sig av *roller* som koncept: en roll definierar ansvarsområdet och beteendet hos en individ eller en grupp av individer:

- Ansvarsområdet definieras genom de *artefakter* som rollen skapar eller modifierar. En roll använder således artefakter för att åstadkomma något, och rollens output blir således en ny artefakt som kan användas av någon annan. En artefakt kan till exempel vara en teknisk modell eller källkoden för en viss funktion.
- Beteendet definieras i termer av *aktiviteter* som rollen utför. De aktiviteter som en roll utför skall vara sammanhängande och passande.

En roll definieras vanligtvis som en hatt, och en projektmedlem kan bära flera hattar. Denna rollfördelning används sedan för att genomföra de nio arbetsflöden som ett RUP-projekt generellt

består av, se figur 5 igen. De tre sista arbetsflödena räknas som hjälpflöden för att hantera risker och förändrade krav samt bistå med planeringsverktyg som stödjer hela processen (Kruchten, 2000).

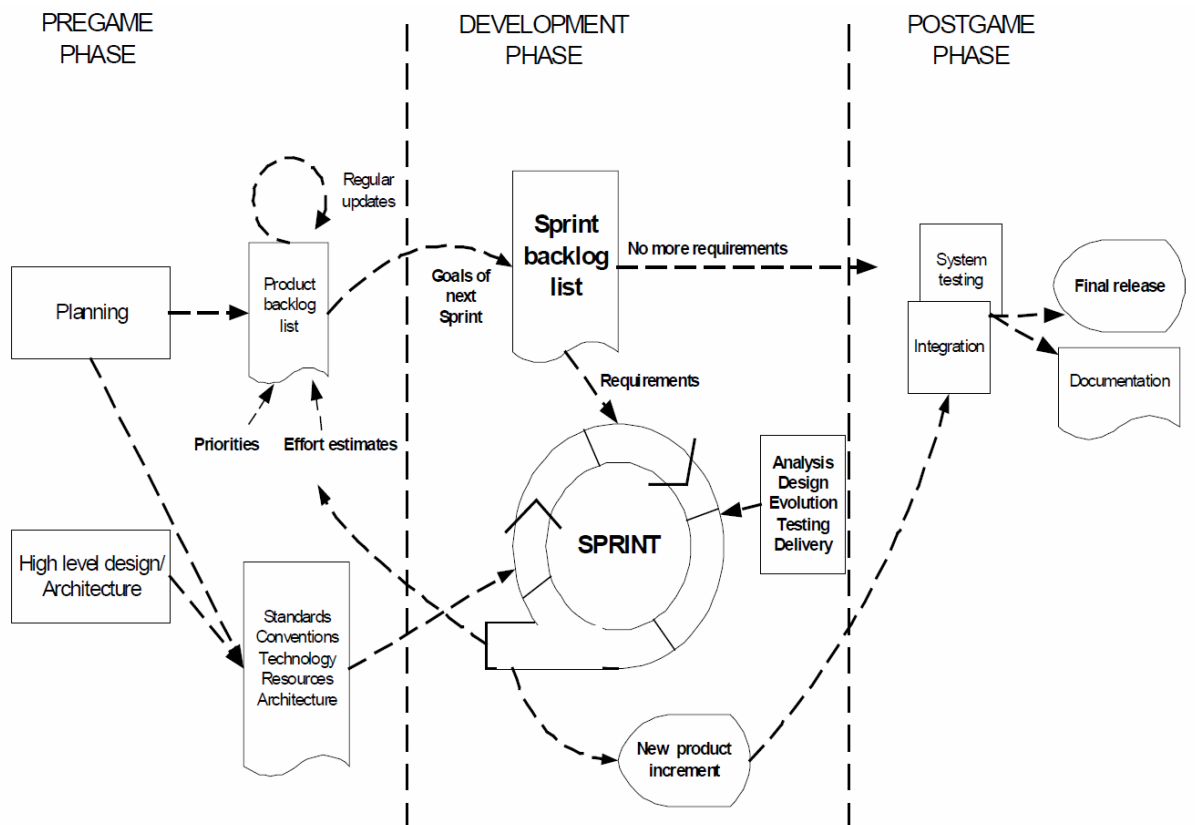
Gibbs (2006) menar att RUP kan vara agilt om utövarna skalar bort de processer och artefakter som ej behövs för att nå målen. Iterationerna skall vara så korta som möjligt, då en agil utvecklingsmetodik förutsätter att testningen sker kontinuerligt. Allt som går att automatisera bör automatiseras, till exempel testningen. När osäkerhet uppstår är det slutkundens ord som skall driva beslutsfattandet. RUP består av över 30 roller, över 20 aktiviteter och över 70 artefakter vilket kräver att användaren av ramverket anpassar det till verksamheten. Ett typiskt problem i detta fall är då den osäkerhet som uppstår när artefakter och roller väljs. Ju fler artefakter och roller som bibehålls i RUP, desto mer normativt blir ramverket vilket ej är förenligt med agila värderingar. Osäkerheten kan leda till en rädsla för att välja bort delar av RUP, vilket leder till att modellen ändå förblir för omfattande för det aktuella projektet (Kniberg & Skarin, 2010).

Omfattningen av RUP-ramverket har gjort att dess lämplighet i mindre projekt har ifrågasatts. I en studie användes RUP-ramverket i ett utvecklingsprojekt med fyra medlemmar, dock i ytterst avskalad form. Studien kom fram till att antalet artefakter som verkligen behövs för ett mindre projekt är tio till tolv stycken, bland annat en utvecklingsplan, ett visionsdokument, en iterationsplan samt en bedömningsmall för varje iteration. Inga formella rollfördelningar skedde i teamet, utan RUP:s rollista användes endast för att säkerställa att all nödvändig kunskap fanns i utvecklingsteamet. Aktiviteter användes inte för att styra arbetet, då erfarna utvecklare ej behöver detaljstyras. Istället lades fokus vid att förtydliga det önskade utfallet av en iteration. Genom att skala av RUP på detta sätt kunde ramverket med framgång användas i mindre projekt med en varaktighet på upp till nio månader. Dock framhävs vikten av kontinuerlig feedback från slutanvändaren för att projektet skall lyckas, då ett av de undersökta fallen misslyckades på grund av den bristande kommunikationen mellan det studerade företaget och dess slutkund (Hirsch, 2002).

3.1.3 Scrum

Scrum är en utvecklingsmetodik som växte fram i Japan på 80-talet som bygger på studier gjorda på tillverkningsindustrin. Denna utvecklingsmetodik anses vara adaptiv, snabb och självorganiserande vilket ansågs vara nödvändigt i ett område där stora förändringar sker kontinuerligt (Abrahamsson et al., 2002). Variabler såsom kundkrav, tidsramar, teknologi och resurser påverkar ständigt mjukvaruutvecklingsförfaranden och kommer sannolikt förändras under tiden, vilket gör utvecklingsprocessen oförutsägbar. Målet blir då att beakta dessa variabler vilket kräver en metodik som är flexibel (Schwaber, 1995).

Scrum bygger på tre faser: *pre-game*, *development* samt *post-game*, se figur 6.



Figur 6: Scrum-modellen och dess tre övergripande faser (Abrahamsson et al., 2002)

Den första fasen består av två underfaser, *planning* samt *high level design*. I planning-fasen skapas en *product backlog*-lista som innehåller alla krav som för tillfället är kända. Dessa krav kan komma från många olika intressenter. Därefter prioriteras kraven efter insatsen som krävs. Product backlog-listan uppdateras kontinuerligt med nya krav eller med mer detaljerade prognoser och vid varje iteration utvärderas listan av teamet för att veta vad som behöver göras vid nästa iteration. I denna underfas definieras också projektteamet, potentiella risker samt de tillgängliga resurserna. I high level design-fasen sker planeringen av hur backlog-posterna skall implementeras. Här diskuteras även inverkan som tekniska utvecklingar har på implementeringen. Uppdateringen av product backlog-listan sker iterativt, men övriga processer i pre-game-fasen sker på ett fördefinierat sätt i likhet med vattenfallsmodellen (Schwaber, 1995).

Den andra fasen är den agila fasen i Scrum. Denna fas beskrivs som en svart låda där oförutsägbara saker kommer att ske. De tidigare nämnda variabelerna följs upp i varje iteration, kallad *Sprint*, och kontrolleras genom exempelvis riskhanteringsförfaranden. En Sprint beräknas ta en till fyra veckor att genomföra, och varje Sprint skall leda till ett förfinande av den utvecklade produkten. Inför varje Sprint skall en *Sprint backlog*-lista upprättas som definierar vad som behöver göras under den aktuella

Sprinten. När denna lista har upprättats sker inga förändringar förrän iterationen har slutförts. Varje Sprint avslutas med en granskning som hela teamet deltar på (Schwaber, 1995).

När teamet anser att variablerna, exempelvis alla krav, har fullföljts går teamet vidare till den sista fasen, avslutningsfasen. Här sker alla förberedelser inför släppet: mjukvaran skall testas, integreras samt dokumenteras innan överlämningen sker. I likhet med den första fasen är även denna fas statisk, och inga nya krav kan ställas (Schwaber, 1995).

I Scrum finns flera identifierbara roller som presenteras nedan. De tre översta är de viktigaste rollerna medan de två sista agerar stödroller (Abrahamsson et al., 2002):

- *Scrum Master*: Denna roll ser till att alla i teamet är fokuserade på arbetsuppgiften och att inga hinder står i vägen för produktiviteten. Denna roll är ej en teamledare då Scrum förutsätter att teamet är självorganiserande. Dock ansvarar denna roll för att projektet arbetar enligt de principer som Scrum förespråkar. Denna roll skall även se till att hålla dagliga Scrum-möten på ca 15 minuter där framfarten och eventuella hinder diskuteras.
- *Produktägare*: Denna roll väljs av Scrum Master samt övriga intressenter och har som uppgift att se till att prioriteringar förstås av alla involverade. Rollen agerar som kundens röst och har ansvaret för product backlog-listan och att den syns och följs. Produktägaren samlar beskrivningar på hur mjukvaran bör fungera och omvandlar dem till tydliga krav på funktionalitet. Denna roll har auktoriteten i teamet.
- *Scrum-teamet*: Teamet har ansvaret för att leverera produkten och består av fem till nio medlemmar. Teamet skall vara självorganiserande och tvärfunktionell och avgör även vilken arbetsinsats som krävs för att genomföra arbetsuppgifterna. Teamet klargör också vilka hinder som måste tas om hand.
- *Intressenter*: Intressenterna bidrar genom kravställning till product backlog-listan och deltar även vid granskningarna.
- *Ledningen*: Ledningen har det slutgiltiga ordet vid beslutssituationer och deltar även då målen och kraven definieras.

Eftersom Scrum i sig inte kräver något särskilt utvecklingspraxis kan den anpassas till att fungera med det tillvägagångssätt som redan används inom en organisation. Den stora skillnaden uppstår främst i de olika rollerna som ett projektteam skall bestå av, då Scrum överlämnar mycket av ansvaret till utvecklingsteamet. Denna utvecklingsmodell lämpar sig således för små team där flexibiliteten, kundens åsikter och anpassningen till förändrade krav är viktiga faktorer (Abrahamsson et al., 2002).

3.2 Lean-principer i mjukvaruutveckling

En filosofi som har fått genomslag inom IT och även på Scania är *lean* mjukvaruutveckling. Denna filosofi bygger på att minimera alla aktiviteter som leder till slöseri och som ej skapar något mervärde i slutprodukten. Lean-filosofin växte fram ur tillverkningsindustrin där Toyota banade vägen för effektivitetshöjande åtgärder. Lean-filosofin finns i ett antal olika tappningar, bland annat lean tillverkning, lean logistikkedjor samt lean produktutveckling. Lean mjukvaruutveckling är en form av den sistnämnda, men där lean-värderingarna appliceras på att effektivisera de processer som ingår vid skapandet av mjukvara, till exempel att ta bort funktionalitet som slutanvändaren ej behöver, en viktig aspekt för mobila appar, samt bibehålla en flexibilitet i planeringen. Gemensamt för alla lean-tappningar är strävan efter att bibehålla värdet mot slutkunden med mindre ansträngning (Poppendieck, 2006).

Lean mjukvaruutveckling bygger på Poppendiecks (2006) sju principer som följer:

- **Eliminera slöseri**

Den viktigaste principen är att eliminera allt slöseri. Allt som ej tillför något värde till kunden kan utvecklingen klara sig utan. För att kunna eliminera slöseri måste utvecklaren till att börja med kunna identifiera slöseriet och ha insikt i vad som anses vara värdeskapande. Detta är dock lättare sagt än gjort. Poppendieck (2006) listar följande punkter som typiska slöseriposter:

- *Halvfärdigt arbete:* Halvfärdiga lösningar tenderar att försvinna, glömmas och dölja kvalitetsproblem. Således gäller det att jobba iterativt och bryta ner arbetet i mindre omgångar. Om flödet leder till en halvfärdig lösning räknas det som slöseri.
- *Extra funktionalitet:* Om det ej finns ett klart ekonomiskt värde i att funktionaliteten utvecklas bör den elimineras. Detta räknas som den värsta slöseriposten och leder lätt till skenande kostnader.
- *Omlärning:* Det är vanligt att tagna beslut glöms längre fram då de ej dokumenteras. Dokumenterade beslut tenderar också att glömmas då de anställda sällan återkopplar till dokumenten. Det är också slöseri att ignorera den kunskap som de anställda tillför. Allt detta leder till att samma kunskap inlärs om och om igen.
- *Överlämning:* När en överlämning till kollegor sker förloras alltid tyst kunskap, särskilt om överlämningsprocessen bygger på att läsa in dokumentation. Målet är alltså att minimera den slöseri som uppstår vid överlämning, till exempel genom att minimera antalet överlämningspunkter och sätta samman tvärfunktionella team.

- *Uppgiftsväxling*: Om en programmerare tvingas till att byta arbetsuppgifter uppstår slöseri, då det är distraherande och leder till att kvaliteten försämras i alla arbetsuppgifter. Varje omställning kräver att den anställde spenderar tid på själva omställningen istället för att arbeta. Ett sätt att motarbeta detta är att klargöra rollerna samt låta de anställda fokusera på en sak i taget.
- *Förseningar*: Mycket slöseri uppstår på grund av förseningar i de olika leden. Tid kan slösas på att försöka ta reda på lösningen till ett problem och förvärras när leveransen från någon projektet är beroende av försenas. Detta kan motverkas av att samla teamet på en plats och jobba i korta iterationer med återkommande feedbacktillfällen.
- *Defekter*: Testning bör ske kontinuerligt för att felen skall åtgärdas. Om en ny typ av fel uppdagas skall framtida testaktiviteter kunna upptäcka den typen. Om mjukvara rutinmässigt når slutstadiet med defekter finns det uppenbara brister någonstans i utvecklingsprocessen som bör åtgärdas snarast.
- *Bygg in kvalitet*: Målet med mjukvaruutvecklingen skall enligt lean vara att bygga in kvalitet direkt. Fokus skall ligga på att defekter ej dyker upp överhuvudtaget under arbetsförloppet. Därmed skall utvecklingsförfarandet ej flytta över kvalitetsansvaret till testningsprocesserna och hoppas på att de upptäcker eventuella problem. Genom att använda korta iterationer med mycket feedback kan defekter snabbt hittas och elimineras.
- **Skapa kunskap**

Mjukvaruutveckling i sig är en kunskapsskapande process, vilket gör det orimligt att skriva kompletta kravspecifikationer vid start. En tidig design kan aldrig förutsäga komplexiteten i själva utvecklingen och den feedback och kunskap som erhålls under resans lopp. Inledningsvis bör endast ett dokument skrivas som förtydligar den funktionalitet som eftersträvas. Upplägget på arbetet bör vara modulärt så att ny funktionalitet kan läggas in vid behov.

- **Senarelägg beslutsfattandet**

Projektdeltagarna skall sträva efter att skjuta upp beslutsfattandet så mycket som möjligt för att försäkra sig om att de har all den information som behövs för att fatta ett klokt beslut. Detta är särskilt viktigt då det är svårt eller omöjligt att vända på ett genomfört beslut. Målet bör dock alltid vara att öppna upp för möjligheten att ändra riktning om det skulle behövas, då ett felaktigt beslut snabbt leder till slöseri.

- **Leverera snabbt**

Kraven från kunden kan förändras många gånger under utvecklingsförloppet, och det är då eftersträfvansvärt att förkorta utvecklingstiden så mycket som möjligt så att de ej hinner ändra sig. Genom att snabba upp processerna minskar risken för att resurser slösas på att ha valt fel väg.

- **Respektera människor**

I en utvecklingsmiljö gäller det att ta vara på den expertis som organisationen besitter. Om organisationen respekterar de anställda kommer naturliga ledare som främjar kreativitet att växa fram. Det gäller dessutom att utveckla expertis inom de områden som företaget vill verka konkurrenskraftigt. Samtidigt skall en tilltro finnas till de anställda och överlåta en stor del av ansvaret, då detaljstyrning oftast hämmar kreativiteten.

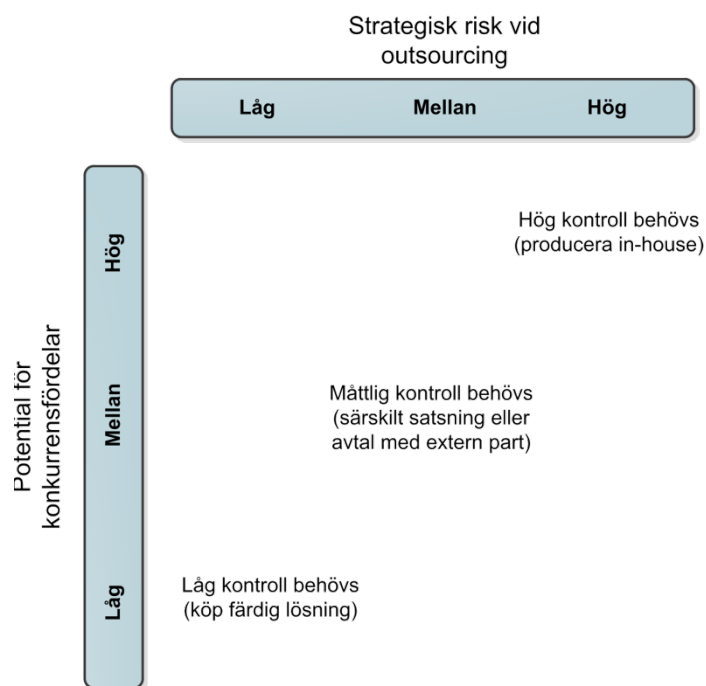
- **Optimera helheten**

En lean utvecklingsmiljö skall optimera hela värdekedjan, från det att ordern har kommit in till att produkten som uppfyller behovet har skapats. Om suboptimering råder, det vill säga endast delar av värdekedjan har optimerats, påverkas hela utvecklingsmiljön negativt. Om hela värdekedjan delar på samma ledningssystem och samma syn på processen kan stora besparingar göras gentemot konkurrenterna.

3.3 Outsourcing av mjukvaruutveckling

Outsourcing, att kontraktera ut en eller flera processer till ett annat företag, är ett vanligt förekommande fenomen. Outsourcing av mjukvara definieras vanligtvis som utläggningen av utveckling, planering, styrning, träning eller underhåll som berör mjukvaruutveckling. Det vanligaste fallet är då utvecklingsprocessen läggs ut och genomförs av utvecklare som besitter den expertkunskap som eftersöks (Abbas et al., 1997).

Vanligtvis brukar en rad olika anledningar existera till varför outsourcing sker. Den vanligaste anledningen är att företaget saknar den kompetens som behövs för att bedriva en ny process på bästa möjliga sätt. Ett företag har vanligtvis en eller flera kärnkompetenser som bör utvecklas till det yttersta för att försöka åstadkomma "best-in-class"-värde åt slutkunderna. Det område organisationen eller avdelningen är bäst på är det område som skall göras internt vilket kräver personalresurser. Quinn (1999) definierar kärnkompetensfrågan som en matris där den vertikala dimensionen representerar potentialen för konkurrenskraft och den horisontella dimensionen representerar den strategiska risken vid outsourcing, se figur 7.



Figur 7: Outsourcing-matris som hjälper beslutsfattare outsourca rätt processer (bygger på Quinn, 1999)

Den vitala kärnverksamheten som knyter direkt till kundbehovet bör utvecklas internt för att bevara kvaliteten och kontrollen. I de fall då ett åtagande klassas som viktigt för att skydda kärnkompetensen, men som nödvändigtvis inte är en del av den, kan en extern part med den rätta profilen för att genomföra åtagandet övervägas.

Undersökningar visar på att företag sällan har förståelse för sin egen kärnkompetens och skulle minimera riskerna och spara resurser på att outsourca delar av verksamheten (Quinn, 1999). Rutinmässiga uppgifter såsom programmering outsourcas ofta, medan vissa specialiserade uppdrag såsom appdesign outsourcas vid behov. Detta behov uppstår då uppgiften i sig inte är en del av kärnkompetensen eller då företaget ej har tid eller möjlighet att införskaffa den kompetens som krävs internt (Abbas et al., 1997). Det har visat sig att företag som försöker rekrytera personal till uppgifter som inte är en del av kärnan sällan lyckas, då de som besitter specialistkunskaper ofta söker sig till företag som redan är ledande inom området. Detta gör att ett företag som outsourcar processer snabbare kommer åt all den kunskap som eftersträvas då expertisen som söks hittas hos företaget som anlitas (Quinn, 2000).

För att ett effektivt utbyte skall kunna ske mellan företaget som outsourcar och de som har åtagit sig uppgiften gäller det att dela på ett tydligt mål och gemensamma principer men att samtidigt ha förståelse för att den andra parten kan arbeta efter andra principer, exempelvis en annan projektmodell (Quinn, 2000). Genom att kombinera olika paradigmer och tillvägagångssätt har det visat sig att lösningarna ofta blir mer genomtänkta då problemet ses med nya ögon och löses på nya, kreativa sätt.

(Abbas et al., 1997). Kontinuerlig feedback och interaktiva tester är däremot av stor vikt vid outsourcing och bör ersätta detaljstyrning för att ej hämma kreativiteten (Quinn, 2000).

En sammanställning av karaktäristiska för- och nackdelar för outsourcing har gjorts av Dibbern et al. (2004), som baseras på ett stort antal fallstudier. När outsourcing sker brukar kostnader reduceras eller åtminstone bli mer förutsägbara då ett tydligt avtal skrivs med motparten. Behovet att anställa specialister minimeras, men samtidigt begränsas långsiktiga karriärmöjligheter internt. Genom att kontraktera ett externt företag får det outsourcande företaget snabbt tillgång till högaktuell teknologi och många specialister samt att utvecklingstakten generellt blir högre. Dock går företaget miste om teknisk kunskap som kan vara till nytta internt. Om ingen kontinuerlig feedback utförs blir kommunikationsproblemen stora: kraven blir svåra att ställa upp och otydliga och det blir även svårt att utvärdera prestationerna och att hålla tidsramarna.

Just bristen på kontroll över utvecklingsförfarandet kan bli ett problem om säkerhet och konfidentialitet är av värde. Det blir då ytterst viktigt att avtalet belyser detta faktum och att det kontrakterade företaget kan försäkra kunden om att kvalitetskraven möts. I kontraktet måste det dessutom tydligt framgå vem som äger rättigheterna för lösningen eller delar av den, särskilt om mjukvaran skall vidareutvecklas eller underhållas (Abbas et al., 1997). En rädsla som också ofta uppstår vid outsourcing är att det outsourcande företags okunskap om området gör att alla förhandlingar och kravsättningen är till leverantörens fördel. Det blir således svårt att ställa konkreta krav när leverantören kan mer än en själv, samt att det blir svårt att estimer framsteg då mätetal blir svåra att definiera för den okunnige.

3.4 Tilläggstjänster för högkostnadsprodukter

Den mobila sfären har blivit ett allt mer intressant område för företag att kliva in i. En av anledningarna kan vara industriföretagens allt större fokus på att sammanväva sina produkt- och tjänsteutbud. Definitionen av en tjänst är bred, men flera forskare hävdar att gränsen mellan en produkt och en tjänst har suddats ut alltjämt. Allteftersom tjänsternas betydelse har ökat har studier observerat att företag alltmer förenar sina produkter med nyutvecklade tjänster som skall erbjuda mervärde till kunden (Johnstone & Wilkinson, 2009).

En industri som ofta omnämns som ett framgångsrikt exempel inom tjänsteutvecklingen är flygindustrin. Då konkurrensen har ökat med åren har produktfokuserade företag blivit tvingade till att på diverse sätt integrera tjänster med sina produkter (Johnstone & Wilkinson, 2009). Flygindustrin kännetecknas bland annat av sin låga produktionstakt, de höga priserna per enhet samt en generellt låg produktoutput, kännetecknen som även delas av företag som sysslar med tunga fordon såsom Scania. I och med att produkterna förväntas ha en lång levnadstid gäller det att finna andra möjliga

inkomstkällor, där attraktiva tilläggstjänster spelar en viktig roll, såsom fordonsservice eller övervakningssystem.

Genom att erbjuda tjänster som konsumenter eftersträvar och som är knutna till de produkter företaget säljer kan nya inkomstkällor och nya differentieringsmöjligheter skapas. Kombinationer av produkter och tjänster kan skapas för att strategiskt bemöta kunders efterfrågan och därmed skapa mervärde på ett mer riktat sätt (Johnstone & Wilkinson, 2009). En fara kan dock uppstå om tjänsten är tätt integrerad med en specifik produkt: kunden kan se tjänsten som en väsentlig del av själva produkterbjudandet och ser därmed ingen poäng i att betala en premie för tjänsten. Det kan således leda till svårigheter att skapa nya inkomstkällor ur tjänsterna som erbjuds, vilket tvingar företaget till att antingen belysa värdet av de högre tjänstenivåerna eller hoppas på att tjänsten bygger upp goodwill för framtiden. Ett sätt att effektivt bygga upp starka relationer till kunderna kan vara att införa en trappstegsmodell, där mer diskreta tjänster finns längst ner och där mer involverande tjänster erbjuds ju högre upp på trappan kunden färdas (Raddats, 2011).

3.5 Användarvänlighet

När skapandet av en app sker måste en förståelse finnas för hur det mänskliga systemet fungerar. Utifrån det skall ett användargränssnitt skapas. Nielsen (1994) har ställt upp tio regler som täcker användarvänlighet i alla användargränssnitt.

- *Överblick av systemstatus:* Systemet borde inom en rimlig tid alltid informera användaren med återkoppling angående vad som sker.
- *Enhetlighet:* När kommunikation mellan användaren och systemet skall ske, skall inga systembaserade termer användas. Upplägget skall följa logiska regler och flöden och även språket skall vara av sådan art som användaren använder i vardagen och därmed kan förstå.
- *Tydliga funktioner:* Funktioner som beskrivs skall vara tydliga. Användare kan av misstag använda sig av systemfunktioner vilket skall vara enkelt att ångra eller avbryta. Ett stöd för ångra och gör om förenklar detta.
- *Enhetlighet och standarder:* Användaren skall inte behöva fundera över om olika termer, situationer och handlingar betyder samma sak. Rekommenderat är att följa plattformsspecifika riktlinjer.
- *Förebygg fel:* Vad som är bättre än ett tydligt felmeddelande är att förebygga att felet uppstår från första början. Arbeta om felbenägna tillstånd eller be användaren om en bekräftelse till vald handling.

- *Förebygg igenkännande före minnesbelastning:* Minimera användarens minnesbelastning av objekt, handlingar och möjliga val genom att göra information synligt om så önskas. Information kring hur systemet skall användas skall visas tydligt eller vara möjliga att ta fram vid behov.
- *Effektiv användning:* Systemet skall vara användbart både för oerfarna och erfarna användare. En sådan anpassning kan göras med till exempel kortkommandon och funktioner till de mest vanliga handlingarna.
- *Estetisk och minimalistisk design:* Dialoger skall inte innehålla överflödigt eller irrelevant information. All överflödigt information i en dialog konkurrerar med den relevanta informationen och minskar därmed dess relativa synlighet.
- *Hjälp användaren att förstå och återhämta sig från fel:* De felmeddelanden som presenteras skall innehålla information i klartext och inga koder. De skall precisera problemet och hjälpa användaren att diagnostisera och lösa problemet.
- *Hjälp och dokumentation:* Trots att det bästa är att användaren kan använda systemet utan hjälp och dokumentation kan detta vara nödvändigt. Sådan information skall vara enkel att hitta, fokusera på användarens uppgift, ha tydliga steg som skall tas och inte vara för omfattande.

3.5.1 Mobila applikationer och användarvänlighet

De mobila appar som utvecklats har gått mot en trend i att ha ett "in-och-ut"-vyflöde. Det kan exempelvis handla om att användaren har telefonboken synlig på skärmen och vill se mer information för en kontakt. Användaren väljer kontakten och går därmed in i en ny vy som visar informationen om den valda kontakten. Användaren har en möjlighet att klicka på användarens adress, om den är ifylld, och får därefter upp en ny vy med en karta där kontaktens adress visas på en karta. Användaren kan välja att gå tillbaka ut ur kartvyn och då hamna i kontaktinformationsvyn igen. Därefter kan användaren gå tillbaka igen och åter hamna i telefonboksvyn.

I Xcode som är utvecklingsmiljön för iOS har Apple implementerat en funktion sen version 4.2 som heter Storyboard. Storyboard används till att skapa ovan beskrivna flöden vilket skapar en bra överblick för utvecklaren. Utvecklaren får visuellt se de olika kopplingarna mellan de olika vyerna som skapats. Det hela ger ett verktyg för att enklare skapa mer användarvänliga gränssnitt (Apple Developer - Cocoa Application Competencies for iOS: Storyboard, 2011).

4 Bakgrund om Scania

Detta kapitel presenterar företaget där studien har genomförts samt den produkt och de processer som ligger till grund för studien. En grundlig presentation av Scantias mjukvaruutvecklingsprocess finns även i detta kapitel.

4.1 Scania-verksamheten

År 1900 bildades Maskinfabriksaktiebolaget Scania i Malmö som tillverkade cyklar. År 1891 bildades Vagnfabriksaktiebolaget Vabis i Södertälje och förkortades till Vabis. Vabis tillverkade järnvägsvagnar. På grund av den växande konkurrensen slogs Vabis och Scania ihop år 1911. Den första bussen tillverkades samma år och därefter har en tillverkning av bilar, bussar och lastbilar fortsatt (Scania - Historia, 2009). Efter ett flertal år placerade Scania-Vabis sin tillverkning i Södertälje. Idag har Scaniakoncernen fler än 35 000 anställda. Forskning och utveckling sker i Södertälje och sysselsätter runt 2900 personer (Scania - Scania in brief, 2011).

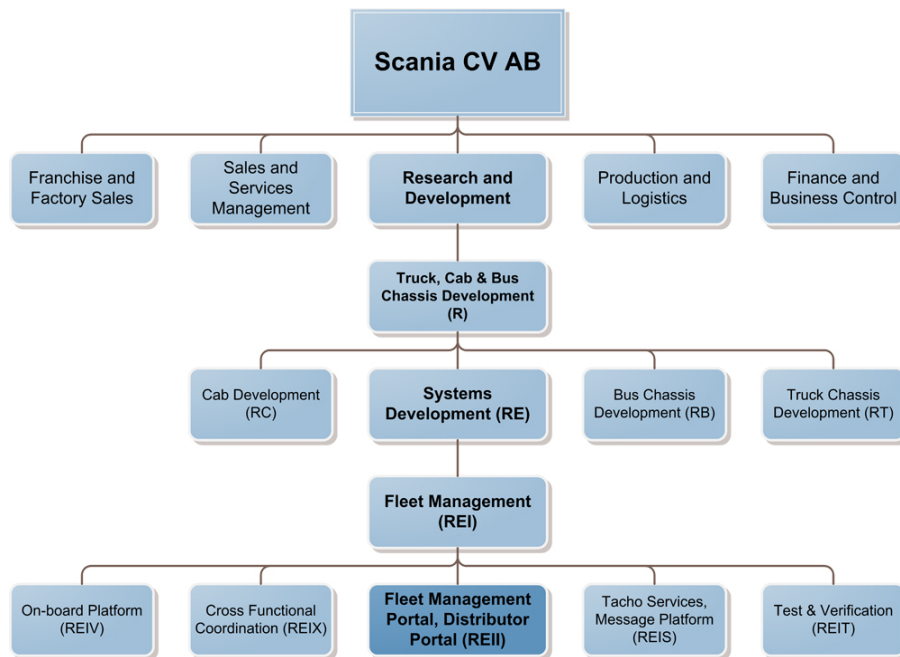
Scantias primära marknad för fordonsleveranser är Europa som står för 39 % av de totala fordonsleveranserna för 2010. Latinamerika, en strategiskt viktig marknad för Scania, står för 32 %. Den asiatiska marknaden, som stod för 19 % av de totala fordonsleveranserna år 2010, spås vara en marknad som fortsätter växa starkt. Den enda del av Scania som närvarar i Nordamerika är Scania Engines, vars fokus ligger på industri- och marinmotorer (Scania årsredovisning 2010, 2011).

4.2 Scania Fleet Management

"Som en naturlig del i Scaniaaffären förstärker vi produkttegenskaperna genom trådlös uppkoppling och ger affärsfördelar till både slutkunder och interna kunder."

(Fleet Management – Mission, Vision & Strategy, 2010)

Scania som organisation är indelat i flertalet sektioner och grupper som inriktar sig på olika områden, se figur 8.



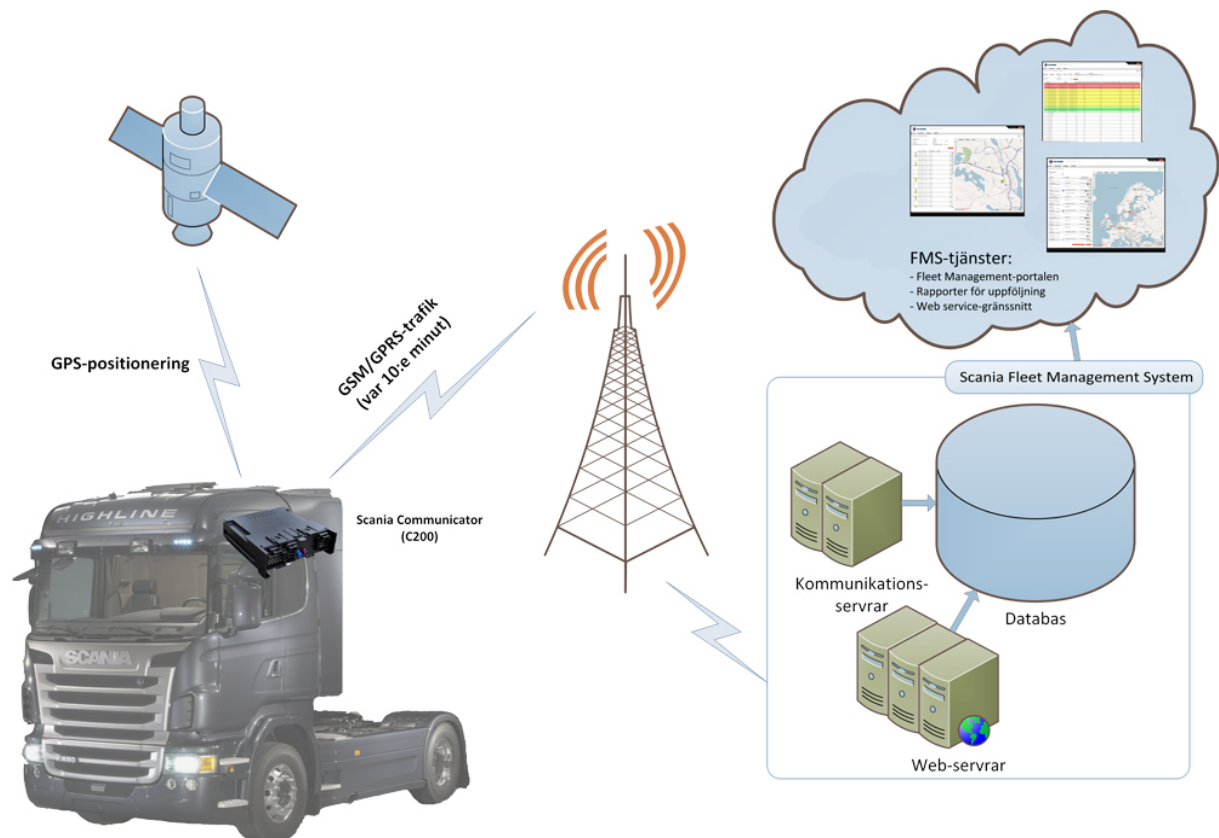
Figur 8: Organisationsschema över Scania och Fleet Management

Scanias forskning och utveckling består av ett flertal underavdelningar varav en av dem är R, som står för “Truck, Cab and Bus Chassis Development”. R i sin tur har en underavdelning vid namn RE, “Systems Development”, där flera former av mjukvaruutveckling sker.

Sektionen för Fleet Management (REI), som består av ca 60 personer, är organiserad i fem olika grupper. Dessa är REII och REIS som hanterar utvecklingen av internetplattformen (FMP) respektive Tachotjänsten, tjänsten som rör de färdskrivare som finns installerade i lastbilar. Gruppen REIV utvecklar Communicator-fordonsdatorn till lastbilen, REIT är en testgrupp och REIX ansvarar för koordineringen och planeringen mellan de ingående avdelningarna och ansvarar även för framtidsstudier och för inhämtning av feedback från både externa och interna parter. Examensarbetet har genomförts i gruppen REII och studiens syfte, att undersöka förutsättningarna för och utmaningarna vid utveckling av appar, bygger på den internetplattform som denna grupp utvecklar och tillhandahåller.

4.2.1 Fleet Management System

I varje lastbil som Scania har tillverkat sedan februari 2011 finns en fordonsdator installerad vid namn Communicator. Det är denna fordonsdator, med nuvarande modellbeteckning C200, som hanterar kommunikationen mot serverna på Scania via GPS-satelliter, SMS- och GPRS-trafik (Scania - Fleet Management, 2009). Därefter kan all insamlad data presenteras i till exempel e-postrapporter eller på webbportalen. Se figur 9 för en illustration över hur kommunikationen ser ut.



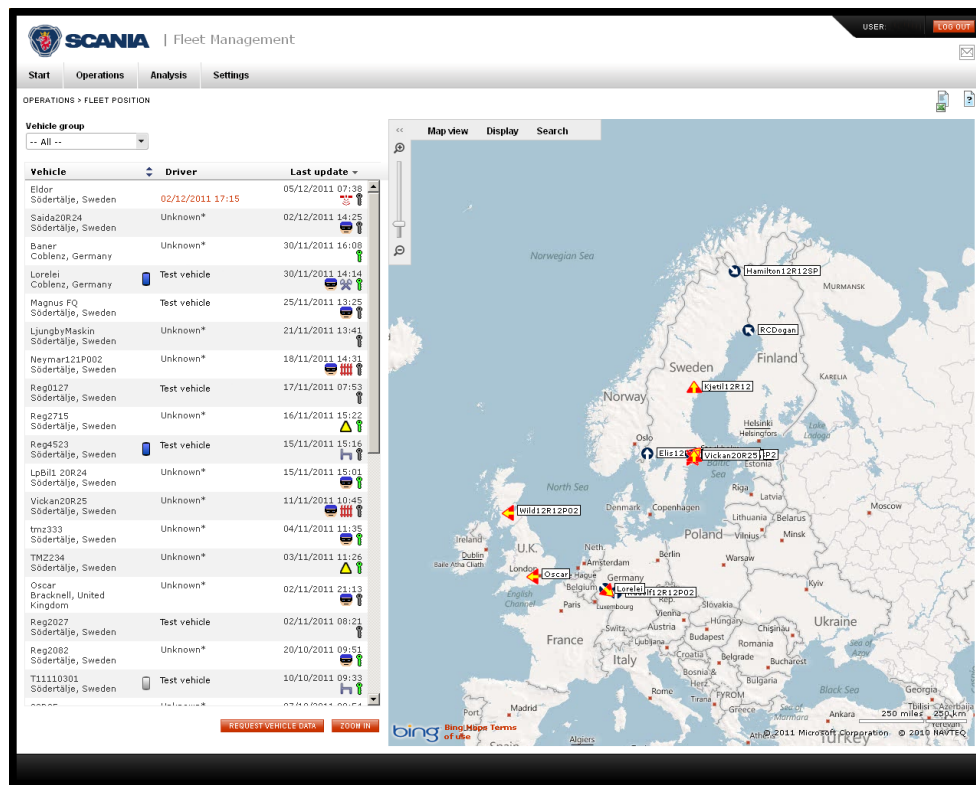
Figur 9: Illustration över hur Fleet Management-datatrafiken ser ut

Hela systemet där data samlas in och som till slut presenteras för kund kallas Fleet Management-systemet, FMS. Genom att Scanias kunder använder sig av FMS skall deras lönsamhet förbättras för tjänster som är relaterade till förare, fordon och enkel transportledning (Fleet Management – Mission, Vision & Strategy, 2010). FMS finns för kund i tre nivåer som omfattar olika grad av kontroll över lastbilarna. Den mest grundläggande nivån kallas Monitoring och erbjuder en funktion där kunden får vecko-, månads- och årsrapporter via e-post. E-posten omfattar rapporter där åkerierna får ta del av bränsleförbrukning, körd avstånd och övriga grundläggande uppgifter för varje fordon. Monitoringtjänsten är standard i alla lastbilar som köps och erbjuds gratis till kunden i fyra år. Vid nästa nivå av FMS-tjänst, Analysis, får kunden tillgång till en analys av historiska data via webbportalen där det går att se statistik på bland annat förar- och fordonsprestanda samt miljö- och felrapporter. Förarna och fordonen klassificeras till tre nivåer efter färgerna på ett trafikljus. Goda resultat ger grönt, gult är ett medelresultat och rött är ett mindre bra resultat. Ett exempelresultat visas i figur 10.

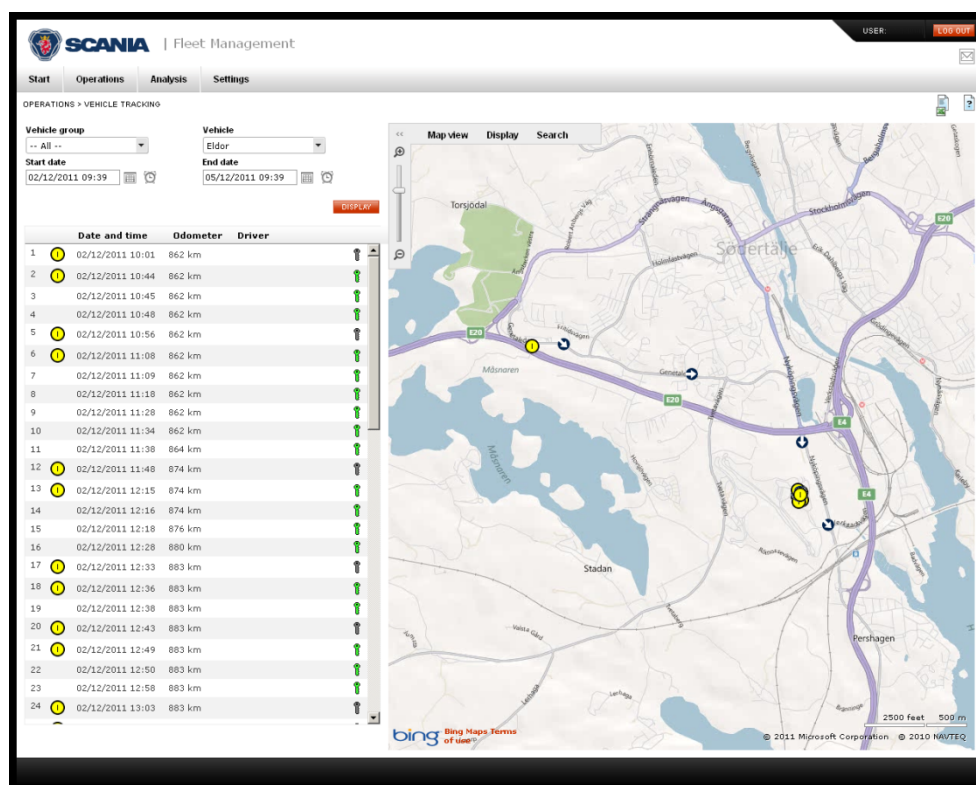
Driver	Start	Stop	Distance (km)	Total fuel consumption (litres)	Average fuel consumption (l/100km)	Average speed (km/h)	Ranking
Unknown*	11/11/2011 09:08	11/11/2011 12:03	63	45	72.2	36.2	16
Unknown*	07/11/2011 13:06	02/12/2011 14:25	1875	789	42.1	37.9	15
Unknown*	25/11/2011 11:15	30/11/2011 14:14	257	110	42.7	60.3	14
Unknown*	15/11/2011 06:26	15/11/2011 14:59	359	158	44.1	60.0	13
Unknown*	30/11/2011 12:32	30/11/2011 15:20	134	42	31.4	48.6	12
Unknown*	22/11/2011 14:33	22/11/2011 15:56	49	20	40.5	61.8	11
Unknown*	18/11/2011 13:06	18/11/2011 14:31	79	22	28.2	56.0	10
Unknown*	09/11/2011 07:36	18/11/2011 08:59	301	88	29.4	66.4	9
Unknown*	14/11/2011 16:48	14/11/2011 21:35	310	125	40.3	70.0	8
Unknown*	29/11/2011 15:01	29/11/2011 15:40	44	17	38.8	67.7	7
Unknown*	22/11/2011 15:57	22/11/2011 16:49	54	20	37.8	62.0	6
Unknown*	22/11/2011 16:55	22/11/2011 20:11	31	12	37.6	59.4	5
Unknown*	10/11/2011 17:12	14/11/2011 13:37	656	254	38.7	71.5	4
Unknown*	18/11/2011 10:21	18/11/2011 13:06	98	31	31.3	59.9	3
Unknown*	22/11/2011 20:11	23/11/2011 06:49	126	46	36.2	70.5	2
Unknown*	21/11/2011 14:03	21/11/2011 16:08	13	4	32.5	37.4	1
Driver1000	NA	NA	NA	NA	NA	NA	0
Driver1003	NA	NA	NA	NA	NA	NA	0
Driver1006	NA	NA	NA	NA	NA	NA	0
Driver1007	NA	NA	NA	NA	NA	NA	0
Driver1008	NA	NA	NA	NA	NA	NA	0
Driver1009	NA	NA	NA	NA	NA	NA	0
Driver1010	NA	NA	NA	NA	NA	NA	0
Driver1011	NA	NA	NA	NA	NA	NA	0
Driver1012	NA	NA	NA	NA	NA	NA	0
Driver1013	NA	NA	NA	NA	NA	NA	0
Driver1017	NA	NA	NA	NA	NA	NA	0
Driver1026	NA	NA	NA	NA	NA	NA	0
Driver1046	NA	NA	NA	NA	NA	NA	0
Driver1055	NA	NA	NA	NA	NA	NA	0

Figur 10: Illustration över en trafikljusrapport i FMS

Med den högsta nivån bland tjänsterna, Control, får kunden även tillgång till övervakningsfunktioner där det går att se var fordonen befinner sig för tillfället på en karta eller historisk positioneringsdata. Figur 11 och 12 visar exempel på detta. Här går det även att lägga ut områden på en karta som kan aktivera notiser eller larm beroende på om bilen åker in eller ut ur zonen, benämnt geofencing.



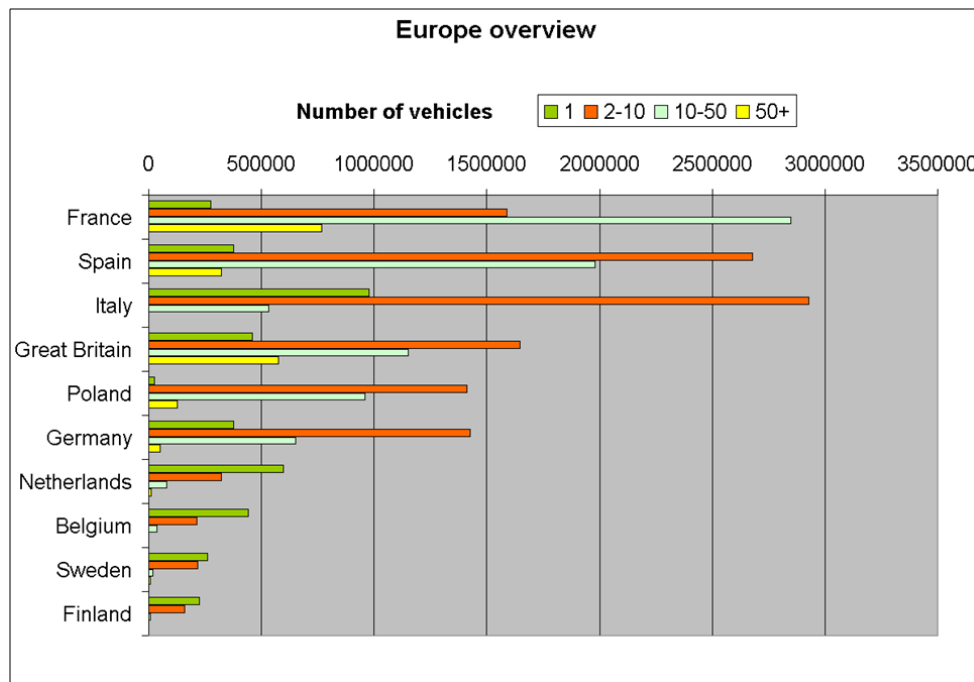
Figur 11: Illustration över kartfunktionen i FMS



Figur 12: Illustration över historiska data i FMS

4.3 Scania Fleet Managements kunder

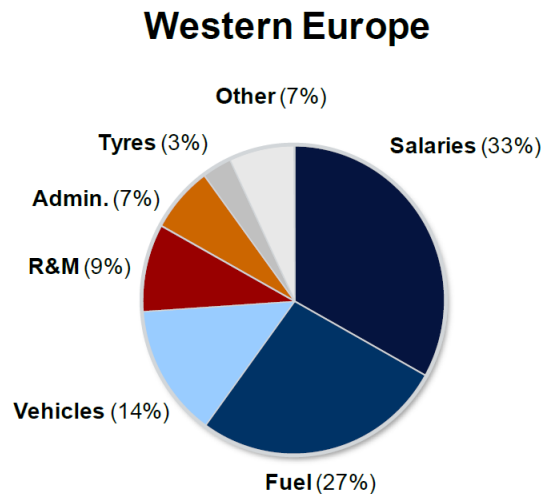
De företag som Scania Fleet Management främst satsar på att nå ut till är de små åkerierna. I denna studie definieras ett litet åkeri som ett åkeri med upp till 20 fordon. I en studie genomförd av Scania har de kartlagt hur fördelningen av tunga fordon ser ut bland åkerier i ett flertal europeiska länder, bland annat Sverige, Frankrike och Nederländerna (Eriksson, 2011). Sammanställningen bygger på siffror från flera källor, där den senaste statistiken är från 2009. Se figur 13 för en illustration över fördelningen.



Figur 13: Översikt över fördelningen av fordon bland åkerier i Europa (Eriksson, 2011)

Bilden illustrerar hur många fordon som är registrerade på ett företag i de europeiska marknaderna, och grupperas efter antalet tunga fordon per åkeri. En majoritet av åkerierna har mellan två och tio tunga fordon, men det finns även en stor grupp av åkerier där fordonsflottan består av mellan 10 och 50 fordon. Antalet åkerier med stora fordonsflottor (50+ fordon) är dock litet i jämförelse.

Anledningen till att Scania fokuserar på de små till medelstora åkerierna är främst för att den stora massan finns där. Konkurrensen bland de väletablerade och stora åkeriverksamheterna är stor, och det blir då fördelaktigt att nå ut till de mindre aktörerna som fortfarande har ett behov av Fleet Management-tjänster. En annan anledning till att diverse Fleet Management-tjänster är attraktiva för transportbranschen är dess förmåga att minimera några av kostnadsposterna såsom bränslekostnader. Figur 14 illustrerar hur kostnadsfördelningen kan se ut för ett genomsnittligt åkeri i västra Europa, och det går att se att bränslekostnaderna tillsammans med förarlönerna utgör de största kostnadsposterna (Eriksson, 2011).



Figur 14: Översikt över typiska kostnadsposter för ett åkeri i Europa (Eriksson, 2011)

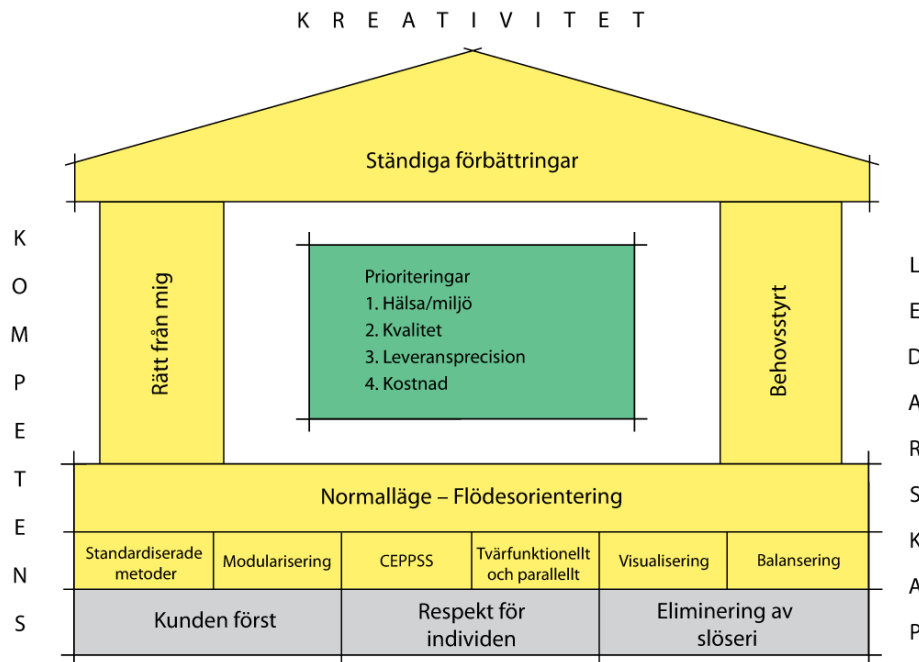
Det finns således en möjlighet till kostnadsbesparande åtgärder både på bränsle- och underhållssidan, och Scania Fleet Management-tjänsten, som ämnar uppmuntra till miljövänligt körande, har som mål att erbjuda detta mervärde. Marginalerna i transportbranschen är dessutom små, vilket gör att alla kostnadsreducerande åtgärder är aktuella. Ur dessa anledningar finns alltså en potential med en mobil tjänst.

4.4 Scanias utvecklingsmodeller

Som ett led i att öka verksamhetens konkurrenskraft har Scanias R&D-gren utvecklat en vision som delas av samtliga underavdelningar. Nästföljande avsnitt presenterar den vision och de utvecklingsmodeller som genomsyrar produkt- och tjänsteutvecklingsarbetet inom R&D.

4.4.1 Scania R&D Factory

För att försäkra den långsiktiga lönsamheten har R&D valt att illustrera Scanias principer, kärnvärden och arbetsmetoder i form av ett Scania-hus, den så kallade *R&D Factory*-modellen. Figur 15 illustrerar detta Scania-hus (R&D Factory, 2010).



Figur 15: Scania R&D Factory som illustrerar Scanias grundläggande kärnvärden (R&D Factory, 2010)

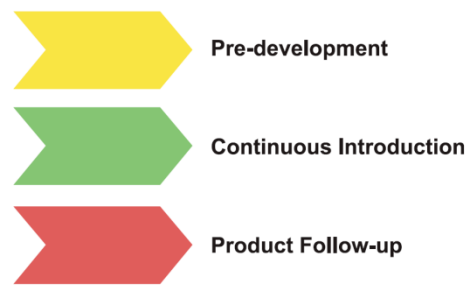
Grunden utgörs av Scanias tre kärnvärden: *kunden först*, *respekt för individen* samt *eliminering av slöseri*. Det första kärnvärdet betonar att kundens behov är i fokus genom hela värdekedjan. Värde skall således alltid skapas ur kundens perspektiv, och Scania behöver därmed agera proaktivt och förutse kundens behov. Det andra kärnvärdet syftar till att organisationen skall se och använda sig av den samlade kunskapen för att förbättra kvaliteten och effektiviteten. Det sista kärnvärdet betonar att allt som ej tillför värde till slutkunden skall elimineras.

Scania utgår ifrån ett normalläge då alla vet hur arbetet skall gå tillväga och hur projekten ligger till i överenskommelsen med slutkunden. Flödesorientering syftar till att allt flödar enligt plan och att optimeringar utförs i de steg där det går, exempelvis förbättrad planering och punktligare leveranser genom hela flödet. Genomströmningen av kunskap och delleranser skall ske på ett smidigt och standardiserat sätt. Detta uppnås bland annat genom att standardisera arbetsmetoder, visualisera projekten så mycket som möjligt samt finna en balans mellan tid och resurser. Detta, tillsammans med husets tre dimensioner, *kompetens*, *kreativitet* och *ledarskap*, ämnar införliva kärnvärdena i allt det arbete som utförs på R&D (R&D Factory, 2010).

4.4.2 Product Development Process

R&D Factory-modellen beskriver R&D:s värdegrund och utgör kärnan i Scanias produktutvecklingsprocess, hädanefter PD-processen. Scanias målsättning är att ta steget från att bara tillverka tunga fordon till att erbjuda heltäckande lösningar, och de kompletterande tjänsternas betydelse kommer därmed fortsättningsvis öka då de bidrar till ökat värde hos produkten samt lägre livscykelkostnader, exempelvis lägre bränsleförbrukning.

Scanias PD-process består av tre delprocesser, *gulpil*, *grönpil* och *rödpil*, se figur 16 (Product Development Process, 2011).



Figur 16: Delprocesserna i Scanias PD-process (Product Development Process, 2011)

Gulpil är en förutvecklingsprocess som består av forskning, teknologiutveckling och konceptutveckling. Denna delprocess ansvarar för inhämtningen av nya idéer och behov, som kan spöras av nya lagar, nya behov hos slutkunderna, nya FoU-framsteg eller av kostnadsrationaliserande åtgärder. Förstudier genomförs som ämnar utforska nya möjligheter och det värde dessa möjligheter kan bidra med, och nya koncept tas fram. När projektet har bedömt huruvida konceptet kan bidra till nyttan, både för kunden och för Scania, väljer deltagarna att antingen gå vidare med idén eller spara den för framtiden. Om deltagarna beslutar sig för att utveckla konceptet sker en övergång till grönpilsfasen.

Grönpil, som Scania kallar för kontinuerlig introduktion, är den huvudsakliga produkt- och tjänsteutvecklingsprocessen. De projekt som når denna stadie kommer med största sannolikhet att leda till produkter och tjänster som erbjuds kunderna. Projekten kännetecknas av att vara planerbara i detalj, och osäkerheten skall vara så låg att projektet följer tidsplanen då kraven på leveransprecision är starka. I grönpilsdelen involveras många fler medarbetare än under gulpilsfasen, som ofta består av små team.

Rödpil är produktuppföljningsprocessen som handlar om att uppdatera och förvalta de produkter och tjänster som används. Avsikten är att förbättra kvaliteten, åtgärda fel som kunden uppdagar samt minimera kostnaderna för alla intressenter. Rödpilsärenden, som kallas för fältkvalitetsuppdrag (Field Quality, FQ), sorteras efter den komplexitet och den tidsram som finns utsatt för att finna en lösning (Product Development Process, 2011).

4.4.3 Software Development Process

Alla projekt på R&D följer PD-processen som diskuterades i föregående avsnitt, men denna process säger inget om hur själva utvecklingsmodellen skall se ut för den mjukvaruutveckling som sker inom organisationen. En process för hur all mjukvaruutveckling skall ske inom Scania har framtagits vid namn Software Development Process (SDP), ett omfattande ramverk som bygger på Rational Unified Process (RUP) (Software Development Process v 3.5, 2011).

Mjukvaruutveckling hos Scania

SD-processen förespråkar explicit att RUP-ramverket skall användas iterativt och inkrementellt samt att utvecklingen skall ske på ett agilt och lean-beaktande sätt (Software Development Process v 3.5, 2011). Ett urval av de roller och artefakter som finns definierade i RUP-ramverket har tagits med i den Scania-specifika versionen, och målet på Fleet Management-sektionen är att representera rollerna så mycket som möjligt (Lundsgård, 2012).

I Scanias implementering av RUP finns ungefär 20 roller, varav hälften klassas som hjälproller. Beroende på projektets omfattning behöver inte alla roller och artefakter som dokumenteras i SD-processen användas, och vissa roller överlappar då. Fyra roller räknas dock som kärnroller och måste finnas representerade inom sektionen:

- *System Analyst*: Står för bland annat användningsfallsmodeller och utformningen av kraven på systemet som bygger på den uppställda affärsmodellen.
- *Project Manager*: Bedriver projektledningen och följer upp framstegen.
- *Test Manager*: Ansvarar för hela testförfarandet och för testplanerna.
- *Software Architect*: Driver de stora tekniska besluten.

Utvecklaren räknas ej som en kärnroll men förutsätts finnas i varje utvecklingsprojekt och blir således en av de viktigaste rollerna. Slutanvändarens roll i PD-processen presenteras ej explicit. Obligatoriska artefakter finns för varje roll, till exempel användningsfallsmodeller som System Analyst-rollen förväntas ta fram. Antalet obligatoriska artefakter hålls dock på en företrädesvis låg nivå, då de flesta artefakter som presenteras är valbara som stöd till projektet. Ett mindre projekt förväntas använda ett mycket mindre urval av dessa artefakter (Software Development Process v 3.5, 2011).

Inom Fleet Management-sektionens undergrupper finns ett antal olika roller som motsvarar rollerna i RUP. På REII finns en objektledare, ett antal kravhanterare som inhämtar information från användarna och utformar användningsfallsmodeller, systemarkitekter som driver de stora tekniska besluten samt en stor grupp utvecklare som består av både fastanställda och inhyrda konsulter. Dessa roller återfinns även i REIS. Objektledaren motsvarar RUP:s projektledarroll, kravhanterarrollen motsvarar System Analyst-rollen i RUP på en FMP-klientnivå, systemarkitekterna motsvarar Software Architect-rollen och utvecklarna Developer-rollen. Det finns också en specialistroll definierad i SDP, Usability Architect, som specialiserar sig i användbarhet hos gränssnitt, i REII:s fall de webbaserade tjänsterna (Software Development Process v 3.5, 2011; Lundsgård, 2012).

På REIT finns ett antal testledare och en stor grupp testare som direkt motsvarar SD-processens testroller. På REIV återfinns systemarkitekter samt utvecklare som ansvarar för utvecklingen av C200-hårdvaran. På REIX återfinns funktionsspecialisterna som motsvarar System Analyst- och Business Analyst-rollerna i RUP på en övergripande tjänstenivå. Business Analyst-rollen ansvarar för att se till

att kraven på systemet faktiskt reflekterar de behov som användarna har (Software Development Process v 3.5, 2011; Lundsgård, 2012). Under Inception-fasen sammanställer och balanserar funktionsspecialisten kraven från externa och interna kunder samt lagkrav till ett paket som även ställs mot tillgänglig tid och resurser. En överlämning av detta paket till kravhanterarna i de andra grupperna sker också under Inception-fasen (Hantering av krav REI, 2010). Se tabell 1 för en sammanfattning av de roller som finns inom Scania Fleet Management.

Roll på Scania	Arbetsuppgift	Motsvarande roll i SD-processen	Finns inom avdelning(ar)...
Objektledare	Projektledare för ett tekniskt objekt	<i>Project Manager</i>	REII, REIS
Kravhanterare	Inhämtar information från användarna, utformar användningsfallsmodeller	<i>System Analyst</i>	REII, REIS
Systemarkitekt	Driver stora tekniska beslut	<i>Software Architect</i>	REII, REIS, REIV
Utvecklare	Står för mjukvaruutvecklingen	<i>Developer</i>	REII, REIS, REIV
Användbarhetsexpert	Specialiserar sig i användbarhet hos gränssnitt	<i>Usability Architect</i>	REII, REIS
Testledare	Ansvarar för testprocesserna	<i>Test Manager</i>	REIT
Testare	Genomför testning av den framtagna mjukvaran	<i>Tester</i>	REIT
Funktionsspecialist	Ansvarar för att se till att systemkraven motsvarar användarnas behov	<i>System Analyst, Business Analyst</i>	REIX

Tabell 1: Sammanfattning av roller inom Scania Fleet Management

Scania SDP är det ramverk som används under hela grönпилsprocessen. När överlämningen av paketet har skett mellan funktionsspecialisten och kravhanteraren börjar kraven utvecklas efter de ramar som den inledande guldilsförstudien har satt upp, och om oklarheter uppstår drivs frågan tillbaka till funktionsspecialisten. Sammanfattningsvis har funktionsspecialisten en stor roll under guldilsprocessen och under Inception-fasen, medan kravhanteraren har en stor betydelse under hela grönпилsprocessen. Kravhantering under grönпил sköts alltså organisatoriskt av till exempel REII och REIV, grupper där tjänsteutvecklingen sker (Hantering av krav REI, 2010). Medlemmarna i Fleet Management-grupperna som inte är kravhanterare eller funktionsspecialister gör kundbesök ytterst sporadiskt och en uppmaning till fler sådana besök finns internt.

Under guldilsprocessen, förstudien som sker innan SD-processen börjar användas, sker arbetet med att förstå behoven genom intervjuer och besök hos kunder. Ett lösningsförslag tas fram och presenteras för slutkunden för utvärdering. Om förundersökningen mynnar ut till ett grönпилprojekt utvecklas

lösningsförslaget vidare till att bli ett mer konkret förslag, och kanske även ett proof-of-concept som visas upp. Förslaget eller prototypen används för att stämma av med kunden att behoven uppfylls, och detta sker ett eller ett par gånger för att kontrollera att projektet fortlöper enligt planen. Mot slutet av ett projekt finns driftprovsfasen där kunden involveras igen och provkör den framtagna produkten. Detta sker innan produkten levereras i Transition-fasen (Goetzinger, 2012).

På Scania Fleet Management är de anställda medvetna om att kraven kommer att förändras under projektets lopp. Målet är dock att hålla nere kravförändringar under Construction-fasen så mycket som möjligt för att ej tappa kontrollen. Dock är det i dagsläget svårt att avgöra hur mycket kravändringar som har skett under projektets gång då kravprocessen på Scania Fleet Management är nyetablerad. Så länge som behovsbilden som sattes upp initialt uppfylls tas ingen ny stor funktionalitet in i ett pågående projekt, då större förändringar även leder till förändringar i uppdragets tid-kostnad-kvalitetkalkyl. Mindre förändringar och förslag som har uppkommit under diskussioner kan dock övervägas om sektionen anser att det är viktigt att få med. (Goetzinger, 2012). Behovsbilden är således relativt statisk under Construction-fasen.

Tidshorisonten för projekt på REII varierar men kan vara allt från tre månader till ett år. På Scania sker fyra produktsläpp per år, men akuta felkorrigeringar behöver ej följa denna planering (Lundsgård, 2011b).

4.5 Scania Fleet Managements konkurrenter

I och med smartphone-industrins höga utvecklingstakt de senaste åren har många organisationer valt att ta steget in i appvärlden. Flera av Scania Fleet Managements närmaste konkurrenter har redan tagit detta steg och utvecklat mobila appar av sina respektive Fleet Management-lösningar. Daimler AG tillhandahåller en Fleet Management-tjänst vid namn FleetBoard (Home - Fleetboard, 2011) som även kan användas via den iOS-app som finns att ladda ner gratis. Även TomTom erbjuder en liknande Fleet Management-tjänst vid namn WEBFLEET. Denna tjänst kan nås via en mobilapp vid namn WEBFLEET Mobile (TomTom WEBFLEET Mobile - Highlights, 2011), som finns till försäljning för Android och iPhone. En tredje tjänst som finns tillgänglig för smartphones är MiX Mobile (MiX Mobile, 2011). Denna mobilapp finns tillgänglig för alla populära mobilplattformar och finns att ladda ner gratis. Ytterligare en aktör är Masternaut, som har en app vid namn Rapide (Apps to Go, 2012) och som finns till iPhone, iPad, Android och Blackberry.

Den stora likheten mellan alla tillgängliga lösningar är att de kräver en aktiv prenumeration av respektive företags Fleet Management-tjänst för att kunna användas. Dessa appar fokuserar på spårning av fordon och presenterar även en del information om fordonen och körningarna. Vissa av lösningarna har funktioner för kommunikation med föraren.

De exempel som har framförts uppvisar tydligt den trend som idag finns. Utöver att företaget erbjuder en tjänst som bidrar med ett mervärde till kunden visar de även att de följer de rådande tekniska trenderna. Därmed behöver inte resultatet i slutändan ge en monetär vinst för företaget utan det kan även skapa en bild utåt som visar på att företaget ligger i framkant av att skapa en affärsnytta genom den tekniska utvecklingen.

5 Apputveckling

I detta kapitel presenteras den empiri som har erhållits vid genomförandet av intervjuerna med benchmark-objekten, med apputvecklare och med åkerier. Inledningsvis diskuteras de organisatoriska förutsättningarna för apputveckling i avsnitt 5.1 för att därefter övergå till att presentera de tekniska utmaningar som råder i dagsläget i avsnitt 5.2 och 5.3. I slutet av varje avsnitt visas en sammanfattning av de intervjuobjekt som har refererats.

5.1 Intervjuer med jämförbara företag

Initialt var det intressant att skapa sig en förståelse över varför företag valde att satsa på appmarknaden. Då appmarknaden är ett relativt nytt område saknade många av benchmark-objekten en djup förståelse för fenomenet, men genomgående är att de undersökta företagen såg en tydligt framväxande apphysteri i och med iPhone-telefonernas framgångssaga. Inledningsvis var behovsbilden otydlig, men när iPhone kom förändrades sättet att presentera redan existerande information. Till skillnad från tidigare mobila lösningar simplificerade iPhone installationsprocessen av appar avsevärt, och företagen noterade att användarnas surfbeteenden förändrades. En generell satsning på mobila lösningar och appar observerades hos andra, och de intervjuade förstod snabbt att området skulle bli hett (Slott, 2011; Andersson F., 2011; Körberg, 2011).

En viktig faktor i anammandet av appar var just att området var så utforskat och spännande. I kombination med den starka framväxten av smartphones började främst visionära medarbetare att titta på potentialen i denna nya marknad. Dock skedde detta vanligtvis i mindre skala då förståelsen för appar inom organisationen varierade och de tillsatta resurserna var ringa. Allteftersom behovsbilden klarnade började förståelsen för appar att öka (Slott, 2011; Nilsson, 2011). Från att ha kontaktats främst i marknadsföringssyfte började apputvecklarnas kunder inse potentialen i mer verksamhetsnära appar och att gammal, kostsam hårdvara kunde ersättas av något mycket billigare (Andersson K., 2011). I Taxi Stockholms fall började appen som ett marknadsföringsprojekt: de ville vara proaktiva och inte låta konkurrenterna komma före. Under projektets gång insåg de dock att appen faktiskt är något som kan vara av betydelse och synen på den ändrades då serviceorganisationens kostnader kunde minimeras om folk började använda appen i större utsträckning än telefonväxeln (Johansson C. V. & Funayama, 2011). I SJ:s fall kom dock tydliga önskemål från kunden om att de ville ha trafikinformation i en app som gjorde att organisationen reagerade (Johansson T., 2011). Avdelningen C på Scania, Corporate Relations, har lanserat en app vid namn Scania Dealer Locator vars syfte är att hjälpa förare hitta till Scania-verkstäder. Projektets grundtanke var att de ville sätta Scania som ett varumärke på appkartan då inga appar från Scania fanns. Känslan var att något behövde göras och appen skulle vara behovsstyrd så att man ej endast sköt från höften (Zandelin, 2011).

Något som flera intervjuade personer har betonat har varit symbolvärdet i att ta fram en app. Genom att ta fram en app kan företaget bemöta kundens behov direkt, och samtidigt visar företaget framfötterna genom att hänga med i den tekniska utvecklingen och utforma tjänster som företags kunder eller medlemmar kan nyttja på ett enkelt sätt. I exempelvis SEB:s fall vill de uppfattas som en modern bank och att de har förståelse för den värld som deras kunder agerar i. Verksamheten visar således att man är med kunden i den aktuella miljön (Resare, 2011; Nilsson, 2011; Lundsgård, 2011b).

Benchmark-objekten delade även med sig av vilka konsekvenser som de lanserade appar har bidragit med. Då apparna har varit förlängningar av redan existerande webbaserade tjänster är det lätt att tro att kannibalisering kommer uppstå, men bland de intervjuade har inga sådana trender uppdagats. I de undersökta fallen har snarare motsatsen gällt: de mobila apparna agerar främst som kompletterande tjänster och har ej haft en negativ inverkan på webbtjänsterna. Man ökar helt enkelt antalet kontaktytor mot sina slutkunder (Slott, 2011; Nilsson, 2011). Om en noterbar ökning av antalet användare har skett har varit svårt att uppskatta, men i Taxi Stockholms fall tror de att appen har lett till en ökad lojalitet bland frekventa resenärer. Dock är lojalitet svårt att mäta, men sammantaget med kostnadsberäkningar har det visat sig vara lönsamt att satsa på en mobil app (Johansson C. V. & Funayama, 2011). I fallet med Sveriges Åkeriföretag, en branschorganisation för svenska åkeriföretag, har dock responsen på appen varit blandad. En förklaring sägs vara att åkeribranschen består av en konservativ skara med få teknikorienterade människor, men även här börjar de se en förändring då fler och fler smartphones används bland medlemmarna (Resare, 2011).

5.1.1 Förankrings- och förvaltningsprocessen

Då appområdet är ett relativt nytt fenomen var det också av intresse att studera hur förankringen av idén gick till hos de jämförbara företagen och vilka förvaltningsproblem som har uppstått. Även här spelar förståelsen för appens nytta in. Generellt har det ej varit ett större problem att förankra idén om apputveckling inom de undersökta företagen, men i vissa fall har budgeten påverkat inställningen till hur fortskridandet har sett ut.

I fallet med Scania Dealer Locator-appen var förankringsprocessen relativt bekymmerslös då avdelningen CC, Communications, är digital av sig. Deras uppdrag är att jobba online via olika typer av kommunikationskanaler, och steget in i appsfären var då ett naturligt. Det kom krav från administrationen om att en Scania-app är ett måste, men behovsbilden behövde förtydligas. Frågan blev då till vem appen skulle rikta sig till och om det fanns ett business case som motiverade investeringen. Ett stort omfång fanns initialt på appen, men då de såg att projektet ej skulle utföras på utsatt tid trättades omfånget ner till att enbart täcka behovsbilden hos företags åkare. Scania Dealer Locator är den första app som Scania har publicerat, och förvaltningen av publiceringsinformationen sköts i dagsläget från avdelningen CC, då det är på denna avdelning alla dokument finns som verifierar att Scania är det företag det utger sig för att vara. Detta är dock endast nödvändigt för en

applansering på Apples App Store, då processen för att publicera en app för Android-telefoner är mycket enklare ur en administrativ synvinkel (Zandelin, 2011). För tillfället har denna Scania-avdelning inloggningsuppgifterna för App Store- och Android-kontona.

På SEB var och är förståelsen för appar relativt låg, och initialt blev det viktigt att sätta budgeten på rätt nivå för att få en signal om att gå vidare. Förankringen skedde ej högt upp och eftersom en osäkerhet fanns drevs idén igenom avsidet av ett fåtal initiativtagare. Den positiva responsen på den lanserade appen gjorde att flera inom organisationen frågade vad mer man kan göra med appar, och en allmän attitydförändring har kunnat urskiljas. Dock finns fortfarande ingen övergripande strategi som inkluderar smartphones, och andra avdelningar som har haft tydliga planer på hur kommunikationen med slutkunden skulle ske blev överraskade när denna nya kommunikationskanal blev populär. Under intervjun betonas det också att initiativtagarna skall vara försiktiga med hur långt förankringen skall ske: det finns många tankar och idéer internt, och om alla skall få genom sin önskan kommer projektet aldrig fortskrida (Nilsson, 2011).

På Sveriges Åkeriföretag såg förankringsprocessen annorlunda ut. Då organisationen är liten var det ej ett problem att förankra appidén med ledningen, men då organisationen är en branschorganisation för svenska åkare behövde idén stämmas av med ett par åkeriföretag för att se om de tyckte att det var intressant. Bortsett från viss investering i hårdvara samt ekonomisk stöd från Trafikverket skedde utvecklingen av appen kostnadsfritt. Om så inte var fallet hade appen förmodligen ej kunnat förankras då resurser ej finns för sådana projekt. Ur förvaltningsavseende har organisationen upprättat ett kostnadseffektivt avtal med en tredjepart för vidareutveckling och felkorrigering av appen, då inget av detta sköts internt (Resare, 2011).

När en app introduceras öppnas en ny kanal för feedback och nya kostnader, eftersom slutanvändarna kommer med synpunkter eller då tekniska förändringar på smartphone-området sker tvingas apputvecklarna anpassa sig. För Svenska Spel-appen kan användaren lämna feedback direkt i appen, via webben eller genom att ringa kundtjänst. Ett veckomöte hålls där kundtjänst deltar och berättar vad som är aktuellt. Ett förvaltningsavtal har upprättats för appen med tredjepart, men koden ägs av Svenska Spel vilket gör det möjligt för dem att ta hem förvaltningen när behovet uppkommer (Körberg, 2011). På SVT, som äger koden för SVT Play-appen, är varje uppdatering av iPhone-appen ett nytt uppdrag mot tredjepart, och en osäkerhet har uppstått kring huruvida det är ett optimalt tillvägagångssätt. Ett supportforum på nätet har införts där slutanvändarna får dela med sig av synpunkter, och dessa synpunkter används därefter och prioriteras i det fortsatta planerings- och förvaltningsarbetet (Slott, 2011). Från apputvecklingsföretagens perspektiv är det en fråga om att komma överens med kunden om hur förvaltningen och uppföljningen skall ske.

Sammanfattningsvis har det varit viktigt att skapa en förståelse för den affärsnytta en app kan bidra med. Om ett tydligt behov finns blir det lättare att få appen att bli en del av ett starkt business case.

Dessutom måste en tydlighet kring förvaltningsarbetet finnas, då en appintroduktion kan ses som ett införande av en ny kommunikationskanal mellan företaget och slutanvändaren. Generellt anser de intervjuade företagen att responsen har varit god, både från slutkunder men även internt från andra avdelningar vars intresse har väckts av apparna. Dock gäller det att grundligt undersöka potentialen hos en app för en viss avdelning, då det kanske ej lämpar sig att introducera en app då kannibalisering kan uppstå (Slott, 2011). Eftersom appar beror så mycket av slutanvändarens synpunkter är det viktigt att upprätta en snabb kommunikationskanal där utvecklarna får ta del av dessa under förbättringsarbetet.

5.1.2 Utvecklingsmodell för apputveckling

Som ett led i undersökningen har benchmark-företagen samt apputvecklingsföretagen undersökts för att ta reda på hur utvecklingsprocessen har sett ut när en app skall skapas. Som tidigare nämnt använder Scania sig av RUP-ramverket för mjukvaruutveckling, och under arbetets gång har det framkommit att alla intervjuobjekt har använt sig av Scrum som projektmodell eller en modifierad variant av den.

Något som samtliga intervjuobjekt har berört är vikten av en iterativ utvecklingsprocess och insamling av feedback, särskilt från slutkunden som appen är ämnad för. Alla apputvecklare har strävat efter att använda Scrum för utvecklingsprocessen eftersom de anser att det är ett modernt och agilt sätt att bedriva mjukvaruutveckling på, men i flera fall har utvecklarna behövt anpassa sig till kundens utvecklingsprocesser samt andra krav och deadlines som gör Scrum mindre lämpligt (Ramberg, 2011; Lundström & Tornblad, 2011; Slott, 2011). Då apputvecklingsprocessen vanligtvis består av små projektteam och där arbetsuppgiftens omfång är greppbart har best practice-metoder och sunt förnuft i kombination med Scrum-principer räckt väl för att genomföra projektet med goda resultat (Lundström & Tornblad, 2011; Zandelin, 2011). De intervjuade apputvecklarna menar också att de bästa resultaten erhålls då medlemmarna i projektteamen får ha specialistroller, exempelvis en specialist för varje mobil plattform, men att medlemmarna ändå måste ha en förståelse för de andras insatser (Ramberg, 2011; Andersson K., 2011). I övrigt ses apputvecklingsprojekt vanligtvis på samma sätt som vilket annat mjukvaruutvecklingsprojekt som helst (Lundström & Tornblad, 2011). En liten skillnad gentemot vanlig mjukvaruutveckling som presenterades är dock det täta samarbetet mellan grafiker, interaktionsdesigners och utvecklare: eftersom dessa roller är så beroende av varandra behöver ett tätt och iterativt samarbete ske mellan dem för att arbetet skall kunna fortskrida (Andersson F., 2011).

Från apputvecklarnas perspektiv är det oerhört viktigt att utgå från kunden då det är de som har slutordet vid beslutssituationer oavsett (Andersson F., 2011). När en kund kontaktar en apputvecklare för att påbörja sitt första appprojekt är mognaden och förståelsen generellt låg, vilket leder till att workshops vid projektstart blir ytterst viktiga. Dessa används för att förstå behovet hos kunden och finna den kärna som eftersträvas i appen. Funktionerna i appen är viktiga att klargöra, men behovet som måste fyllas är viktigare att följa (Lundström & Tornblad, 2011; Zandelin, 2011). En workshop kan ta upp till åtta timmar att genomföra och används för att samla in alla tankar kunden har samt öka deras förståelse för vad en app faktiskt innebär. Workshopen leder till ett koncept som presenteras för kunden och om konceptet behöver förfinas leder detta till en iterativ process tills parterna har kommit överens. Konceptet används därefter för att skapa en kravspecifikation enligt Scrum-metodiken, med användningsfallsbeskrivningar, backlog-listor och sprintar. Redan under denna stadie kan utvecklarna se om projektets omfattning blir för stor, vilket kan göra det lämpligt att dela upp projektet i delprojekt för att finna en lämplig avgränsning för den eftersökta funktionaliteten (Andersson F., 2011). Om det passar kunden brukar Sprint-längden hållas kort, till exempel två veckor, för att uppmuntra till många avstämningar och iterationer. Avstämningar blir särskilt viktiga om den tekniska lösningen är beroende av många intressenter, och fordonsindustrin gavs som exempel där många tekniska och lagliga faktorer kan påverka om appen exempelvis skall kunna fjärrlåsa ett fordon (Nilsson, 2011; Ramberg, 2011). Kontinuerliga avstämningar och många iterationer är också det mest effektiva sättet att införliva de krav och idéer som uppkommer från kunden och slutanvändarna och har betonats från flera parter. En tanke som dock kom upp var att om appen och dess betydelse blir mer komplicerad framöver kan det vara bättre att ta en titt på ett heltäckande ramverk som exempelvis RUP för att kunna handskas med stora externa beroenden (Nilsson, 2011).

Ett typiskt problem som har dykt upp är att utvecklare försöker omvandla backlog-listan i Scrum till en tollgate-lista istället, vilket leder till att eftersträfvade funktioner prioriteras fel och att förväntningarna på appens omfång är feljusterade, då en tollgate-lista till skillnad från en backlog-lista menar att allt på listan måste implementeras. Det blir således viktigt att förstå vad kunden verkligen vill ha och prioritera bland dessa krav (Ramberg, 2011). En rekommendation är att appens omfång bör trattas ned till att direkt beröra behovsbilden för att kunna lansera appen inom en rimlig tidsram (Zandelin, 2011; Ramberg, 2011). Detta leder också till att backlog-listan blir mer hanterbar. På Svenska Spel genomförs omfattande användartester så att utvecklarna ej utgår från vad de själva tycker när de sedan utvecklar. Detta tankesätt leder till många iterationer där feedback från testanvändare samlas in (Körberg, 2011). För SVT Play-appen har SVT satt upp ett supportforum där användare kan lämna feedback som sedan analyseras internt och används inför den fortsatta utvecklingen av apparna (Slott, 2011).

När det gäller tidsramen har genomsnittstiden hamnat på runt tre till sex månader för ett helt apputvecklingsprojekt, men benchmark-företagen anser att apputvecklingsprojekten generellt har tagit

längre tid än de planerade. Kvartalsvisa släpp för större funktionalitetsbringande uppdateringar är en god riktlinje, samt att felkorrigeringar skall kunna släppas så fort som möjligt (Nilsson, 2011). Internt har processerna uppfattats som tungrodda och byråkratiska, och i fallet med Scania Dealer Locator ville avdelningen vara snabbfotad vilket var den stora anledningen till att appen utvecklades av en tredjepart efter en lös best practice-modell (Zandelin, 2011).

Att projekten drar ut på tiden sker vanligtvis om komplexiteten är hög vilket den blir om visionen och kravställningen är otydlig, något som framkommer när en kund vill skapa sin första app (Lundström & Tornblad, 2011). Dessutom inverkar ofta integrationsarbetet på tidshorisonten, eftersom utvecklarna då blir beroende av andra, till exempel företagets IT-avdelning som kanske ej har mycket resurser för att tillgodose tredjepartens behov av information. Ett integrationsarbete som kräver förändringar i existerande system kan också förlänga tidsramen markant då dessa ofta leder till större interna projekt (Lundström & Tornblad, 2011; Nilsson, 2011). Komplexiteten blir ännu högre om apputvecklingen är beroende av andra organisatoriska processer eller annan teknik. Då måste apputvecklingsprojektet synkroniseras med de andra processernas fortskridande, vilket kan vålla störningar (Ramberg, 2011).

Beroende på appens komplexitet kan längden på själva utvecklingen variera kraftigt, men det kan även vara den inledande kravställningen som tar tid, särskilt om förståelsen för behovsbilden är låg. Ett exempel som framfördes var att den effektiva utvecklingstiden för Scania Dealer Locator-appen endast tog ett par veckor, men om feedbackrundor och förändringar i specifikationerna och designen räknas med tog förfarandet några månader. Att komma på vad man vill göra och hur det skall göras tar således väldigt lång tid i sig (Andersson K., 2011; Andersson F., 2011). Under planeringsfasen kan dock app-prototyper till stor fördel användas, då det är lättare att visa hur utvecklaren har tänkt sig att en lösning skall fungera än att ständigt fråga (Andersson F., 2011). Dock rekommenderar intervjuobjekten att den inledande planeringen skall vara löst sammansatt, då området rör sig snabbt. Som exempel kan nämnas att SVT undviker att ha stora och svåröverblickliga lanseringar, samt att allt som ligger längre än tre månader framåt räknas som en osäkerhet som kommer förändras under arbetets gång (Johansson T., 2011; Zandelin, 2011; Slott, 2011). En annan rekommendation som dök upp var att en första version bör lanseras snabbt så att utvecklarna och kravhanterarna erhåller feedback från de faktiska användarna. Denna feedback kan därefter användas för att ta ställning till vad som skall vara med i nästa version (Andersson K., 2011).

Sammanfattningsvis har det varit viktigt, både för apputvecklarna och för benchmark-företagen, att lyssna på varandra och klarlägga behovsbilden och vad företaget vill att appen skall göra och Scrum har varit ett starkt agilt verktyg för att genomföra detta.

5.1.3 Intern respektive extern utveckling

En annan stor del i denna studie har varit att förstå varför jämförbara företag har valt att antingen utveckla appen internt, så kallad in-houseutveckling, eller lägga ut arbetet på en tredjepart, det vill säga outsourcing. Under arbetets gång stod det tydligt att samtliga benchmark-företag som ställde upp i studien valde att anlita externa apputvecklare för framtagningen av deras appar. I ett par av fallen visade det sig att all IT-relaterad verksamhet läggs ut på tredjepart, det vill säga ingen form av utveckling sker inom företaget. I Scania Fleet Managements fall sker dock all kritisk mjukvaruutveckling internt.

Den gemensamma faktorn som samtliga företag har angett som skäl till outsourcing är att de ej har haft kompetensen för att utveckla en app, åtminstone inledningsvis. I Svenska Spels fall är all utveckling som har att göra med transaktioner av olika slag gjord internt då det är deras kärnverksamhet. Apputvecklingen lades ut främst för att företaget ville få in den externa resursens kunskaper om design och användarupplevelser. De intervjuade anser att för att resultatet skall bli riktigt bra skall utvecklaren ha utvecklat appar innan: utbudet av företag med duktiga apputvecklare är stort, och ju rikare användarupplevelsen behöver vara, desto mer lämpligt blir det att outsource utvecklingen (Körberg, 2011). I Scania Dealer Locator-fallet behövde de känna sig trygga i att jobba med leverantörer som har levererat appar för globala företag sedan tidigare (Zandelin, 2011). Ett mål för Svenska Spel har varit att hitta en leverantör som kan genomföra alla de steg som förväntas i en komplett utvecklingsprocess, till exempel idé, design, bakgrundsflöden, användartester och prototyper. Det har dock visat sig vara svårt då många apputvecklingsföretag ej har den erfarenhet som krävs för att driva igenom alla steg i en sådan process (Körberg, 2011).

En annan anledning till att de jämförbara företagen har gått till en tredjepart är för att erhålla den spetskompetens som apputvecklingsföretagen huserar i ett starkt föränderligt teknikområde. I SEB:s fall uppskattas tredjeparternas agila tillvägagångssätt, då appområdet rör sig fort. Som ett exempel kan en uppdatering av mjukvaran i en telefon föra med sig oväntade konsekvenser, och då gäller det att ha kompetens som förstår innebörden av förändringen. Om företaget besitter en begränsad omvärldsbild eller ett icke-agilt tänk internt går det ej att säkerställa att alla parametrar viktiga för en framgångsrik lansering beaktas (Nilsson, 2011). I Taxi Stockholms fall menas att leverantörens marknadskompetens är oumbärlig och att det är något som de själva skulle ha svårt att hålla sig uppdaterade inom. När den tekniska utvecklingen går så snabbt som den gör behövs den färska kompetensen (Johansson C. V. & Funayama, 2011).

Under intervjuerna har det betonats att förarbeten och mjukvaruutveckling kräver olika kunskaper och att detta bör ses ur två dimensioner: ett förarbete kräver en god kreativ höjd medan utvecklingen kräver en god leveransförmåga och förvaltningskunskap. Att besitta båda dessa dimensioner är svårt för ett företag vars kärnkompetens ej behandlar dessa områden explicit (Körberg, 2011). Det viktigaste

är dock att ständig återkoppling sker mellan leverantören och kunden och att parterna har en samförståelse (Andersson K., 2011). Generellt menar de intervjuade att tredjepartsleverantörer har exakt den spetskompetens de eftersöker inom bland annat användarvänlighet och interaktionsdesign och att det då blir lätt att nyttja denna kompetens. Dessutom är en leverantör som är insatt i apputveckling fullt medveten om de förändringar som sker inom området.

Under intervjuerna med apputvecklingsföretagen framkom det att de även kan hjälpa ett företag med att sätta upp en utvecklingsorganisation själva, men att sådana förfrågningar ej har skett hittills. Eftersom den teknologiska utvecklingen inom appområdet sker snabbt är det svårt för apputvecklingsföretagen att personal som besitter den kompetens som eftersökes. En apputvecklare kände att de har det svårt nog att finna spetskompetens själva, vilket gör att det kan bli svårt att hjälpa andra att hitta resurser. Detta blir särskilt besvärligt om en app behöver lanseras snabbt eller om utveckling till flera plattformar sker, då specialister för varje plattform kan behövas (Andersson F., 2011).

Något som har belysts under intervjuerna är den avsaknad av kontroll som uppstår om de låter leverantören stå för utvecklingen och förvaltningen av appen, särskilt om appen visar sig bli en viktig del av ens tjänsteutbud. När ett företag kommer till en apputvecklare finns ofta en avsaknad av kunskap och förståelse för ämnesområdet vilket gör att det är svårt att ställa krav (Zandelin, 2011). Inställningen är att man bara vill ha en app, men förståelsen för det behov som skall uppfyllas är låg. Ofta är det inte ens de IT-kunniga avdelningarna som kommer med uppdraget, utan det kan vara en marknadsavdelning som initierar idén (Lundström & Tornblad, 2011).

Om apputvecklingen sker internt erhålls fullständig kontroll över bland annat kostnader och den tekniska lösningen och företaget säkerställer att kunskap och kompetens införlivas inom organisationen. Om rutinmässiga förvaltnings- och arbetsuppgifter kan göras in-house, till exempel felkorrigering av appen, kan det bli billigare i längden än att ständigt behöva förlita sig på en tredjepart. Dock finns en risk med att det blir kostsamt att utbilda personal internt samt att det kan vara svårt att finna tillräckligt med arbetssysslor för att motivera en intern utvecklare. Dessutom finns det en risk att spetsen tappas om utvecklingen sker internt, men då är frågan om organisationen alltid måste vara i spetsen för att kunna erbjuda en användbar mobiltjänst (Nilsson, 2011). Ett av målen hos SVT är att finna en lösning för att kunna lansera appar till flera olika mobila plattformar, och då är tanken att en lös konsult kan plockas in för att lansera en plattformsversion av appen snabbt. Detta gör att problematiken med kostsamma kontrakt med en extern leverantör kan undvikas (Slott, 2011). På SEB menar man att om apputveckling skall ske in-house bör ett centralt organ stå för apputvecklingen så att så stora delar av organisationen som möjligt kan ta hjälp av organet vid behov (Nilsson, 2011). För att motivera utveckling in-house behövs en tydlig strategi: om strategin är oklar är det förmodligen bättre att lämna över uppdraget till en tredjepart för att minimera risken (Resare, 2011). Ur Svenska

Spels perspektiv kan in-house-utveckling endast motiveras om kvaliteten på appen blir bättre eller om hastigheten till marknaden med all säkerhet ökar (Körberg, 2011).

Ytterligare en fråga som har diskuterats är äganderätten av koden. De intervjuade apputvecklarna har olika åsikter om detta: antingen kan kunden ta över äganderätten av koden vilket låter dem fortskrida fritt, eller så har kunden enbart brukanderätt vilket gör att endast apputvecklaren har tillåtelse att modifiera koden för mjukvaran (Andersson K., 2011; Ramberg, 2011; Andersson F., 2011; Lundström & Tornblad, 2011). Apputvecklarna har förståelse för att kunden kan vilja förvalta produkten själv framöver och att utvecklaren är med i en eller två versioner av appen i början (Ramberg, 2011). Denna typ av situation förutsätter att kunden äger koden och har rätt att modifiera den framöver.

Ett problem som ofta förekommer vid outsourcing av apputvecklingen är integrationsarbetet. De intervjuade apputvecklarna nämner att tidsfördröjningar sker då de behöver integrera appen till existerande bakomliggande system hos kunden. Det beror dels på att ett sådant arbete tar tid i anspråk från kundens IT-ansvariga, dels på att systemen kanske ej är utformade på det sätt som appen kräver eller på det sätt som apputvecklarna förväntar sig. Ett typiskt problem som nämndes av flera parter var problematiken som uppkom då information behövde extraheras från kundens system (Lundström & Tornblad, 2011; Andersson F., 2011; Ramberg, 2011; Zandelin, 2011).

Sammanfattningsvis har de jämförbara företagen värdesatt den flexibilitet som agila externa parter har medfört. Spetskompetensen inom och den goda förståelsen för appområdet finns naturligt samlad i dessa kanaler vilket ofta är en anledning till att outsourca. Det är dock värt att upprepa att graden av IT-verksamhet har varierat bland de undersökta företagen och att i de fall där ingen IT-utveckling bedrivs internt så är det givet att även apputvecklingen skall outsourceas.

5.1.4 Användarvänlighet och slutanvändarens roll

Något som har betonats i all kontakt med de intervjuade personerna har varit att slutanvändarens åsikter om hur systemet, i det här fallet appen, skall fungera. Många utvecklare gör misstaget att tro sig veta vad användaren vill ha men där åsikterna inte överensstämmer med användarens (Körberg, 2011). Därmed är det viktigt att få in åsikter om hur slutanvändaren skulle vilja se att appen fungerar och vartefter utvecklingen sker skall återkoppling ske kontinuerligt. Användartester ger en kunskap som är svår att få in från andra kanaler och är unikt för varje enskilt fall. Att kombinera de åsikter som användarna har med den kunskap som finns inom interaktionsdesign kommer ge det bästa resultatet med en rik upplevelse som är smidigt och enkelt att använda. De bästa apparna är de enkla apparna. Användaren skall inte behöva support då funktionaliteten skall vara självklar (Resare, 2011). Därmed är det även att föredra att användaren känner igen sig i miljöer som är specifika för den använda plattformen: iPhone-användare vill känna igen sig i iPhone-miljön och Android-användare vill känna igen sig i Android-miljön (Ramberg, 2011; Johansson C. V. & Funayama, 2011) och all interaktion mellan användaren och enheten skall vara naturlig.

Då en app har publicerats är det vanligt att utvecklare vill implementera fler funktioner i appen. Det är även viktigt att denna nya funktionalitet inte försämrar användarvänligheten vilket i sig skulle göra att helhetsresultatet försämras (Johansson C. V. & Funayama, 2011). Den återkoppling som borde samlas in efter färdig produkt kan komma från vitt skilda grupper och därmed ha väldigt olika åsikter. Det gör att företaget ibland måste prioritera och överväga vissa förändringar. Det gäller då att fokusera på de viktigaste målgrupperna (Johansson C. V. & Funayama, 2011) och prioritera den gruppens önskemål högst.

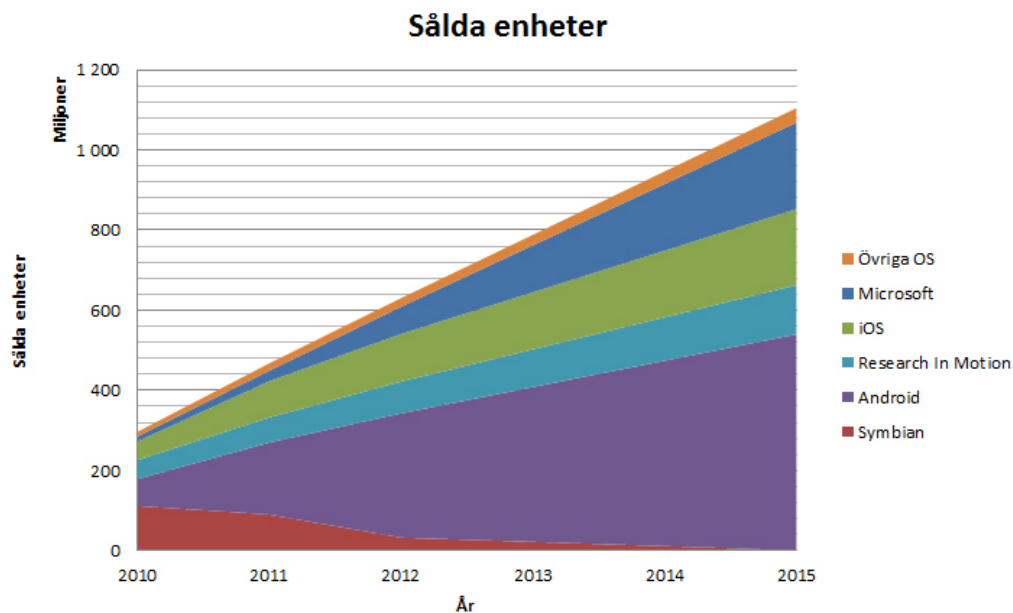
Tabell 2 visar en sammanfattning över de refererade intervjuobjekten i detta avsnitt.

Intervjuobjekt	Företag
Anders Lundsgård	<i>Scania Fleet Management</i>
Carina V. Johansson & Tina Funayama	<i>Taxi Stockholm</i>
Elin Lundström & Rickard Tornblad	<i>Decuria</i>
Erica Zandelin	<i>Scania Corporate Relations</i>
Erik Ramberg et al.	<i>SWAD (numera Springworks)</i>
Fredrik Andersson	<i>Kondensator</i>
Gunnar Slott	<i>SVT</i>
Ingemar Resare	<i>Sveriges Åkeriföretag</i>
Kenneth Andersson	<i>The Mobile Life</i>
Marcus Nilsson	<i>SEB</i>
Patrik Körberg	<i>Svenska Spel</i>
Tobias Johansson	<i>SJ</i>

Tabell 2: Sammanfattning av intervjuobjekt samt företag

5.2 Utveckling till mobila plattformar

Idag är mobiltelefonerna avancerade och har funktionalitet motsvarande en dator. Analysföretaget Gartner förutspår att försäljningen av smartphones kommer att passera försäljningen av PC-datorer vid slutet av år 2011 (Petty & Stevens, 2011b). Figur 17 visar den totala försäljningen hittills och Gartners prognos för respektive mobila plattform.



Figur 17: Prognos för antalet sålda smartphone-enhet per plattform (bygger på Pettet & Holley, 2011b).
Notera att år 2013 och 2014 är egna uppskattningar

Den breda funktionaliteten och stora spridningen av smartphones gör att affärsmöjligheterna även växer. Smartphones har avancerade funktioner som GPS, WiFi-anslutning och övrig hårdvara som ger möjligheter som tidigare inte funnits. De största plattformarna i dag är Android och iOS, men flera andra plattformar finns i periferin. Det som alla plattformar har gemensamt är att vid utvecklingen till respektive plattform binder utvecklaren sig till plattformsägarnas villkor och förändringar (Slott, 2011). Det medför att alla förändringar som ägarna av plattformarna gör måste följas för att de egna apparna skall fungera. Därmed måste de lanserade apparna alltid testas med de nya versionerna av plattformar som släpps.

5.2.1 Utveckling till iOS

Operativsystemet iOS är utvecklat av Apple och lanserades i juni år 2007 (Honan, 2007). iOS släpptes i samband med telefonen iPhone och finns idag även på iPad och iPod touch (Apple Developer - Develop on your Mac, 2012). När iPhone lanserades var den unik med sin stora rörelsekänsliga skärm och avsaknaden av fysiskt tangentbord. Användaren kunde interagera med den så kallade touchskärmen genom olika rörelser av fingrarna (West & Mace, 2009).

Utvecklingen till iOS sker mot iOS SDK i utvecklingsmiljön Xcode som är ett IDE utvecklat av Apple. Med hjälp av Xcode skapas de användargränssnitt som eftersträvas och däri skrivs även koden som ligger bakom användargränssnittet (Apple Developer - Tools Workflow Guide for iOS: About the Tools Workflow for iOS, 2012). Med Xcode fås därmed alla utvecklingsverktyg för att utveckla appar till Apple-produkter. I Xcode ingår Xcode IDE, Instruments och iOS Simulator. Koden skrivs i programmeringsspråket Objective-C (Apple Developer - iOS Overview, 2010). Utvecklingen till iOS

måste ske på en Mac-dator då Xcode endast finns tillgängliga för Mac OS X (Apple Developer - iOS Starting Point, 2011).

5.2.2 Utveckling till Android

År 2007 bildades Open Handset Alliance av Google som är en allians av mobilföretag som driver på att Android skall bli ett världsledande operativsystem. Det gjorde att Google kunde implementera sin egen söksida i operativsystemet, precis som Internet Explorer finns i Windows (Hall & Anderson, 2009). Android är en modifierad version av Google-kärnan och är ett operativsystem med öppen källkod vilket gör det gratis. Programmeringsspråket som används för att utveckla till Android är Java och Javakoden konverteras till Dalvik vid körning. Dalvik är den virtuella maskin som är optimerad för mobila telefoner för smidigast körning (Android Developers - What is Android?, 2012).

För att utveckla till Android används i huvudsak programmeringsspråken Java i kombination med XML. Android har även stöd för C och C++ som kan användas för att utveckla vissa komponenter. Vid utveckling finns tillgång till stora delar av telefonens hårdvara och samma API som grundapplikationerna i telefonen använder sig av (Android Developers - What is Android?, 2012).

För att sätta upp en utvecklingsmiljö till Android måste ett antal steg utföras. Dessa steg finns bäst beskrivet hos Android (Android Developers - Installing the SDK, 2012) men presenteras även här i korthet. Vid utveckling rekommenderas Eclipse men det går även att skriva koden i ett annat IDE eller direkt i en textredigerare. Eclipse rekommenderas för att det där direkt går att anropa de andra resurser som krävs för att utveckla en app (Android Developers - Introduction, 2012) och rekommenderat är även att ADT (Android Developer Tools)-pluginet används. Till Android går det att utveckla i antingen Windows, Mac OS X eller Linux (Android Developers - System Requirements, 2011). Då utvecklingen sker i programmeringsspråket Java krävs även att Java Development Kit finns installerat på datorn. Även Android SDK måste installeras som är de mest grundläggande verktygen för att arbeta med Android. Därefter måste ett SDK installeras för en specifik Android-plattform. Valet av plattform beror på till vilken version av Android utvecklaren vill utveckla sin app men minst en plattform måste installeras. Med plattformen följer även ett emulatorskal som det går att testa sin app på. Det går även att testa sin app direkt i en telefon som kör Android. Görs utvecklingen på Windows krävs USB-drivrutiner för detta (Android Developers - Installing the SDK, 2012).

Idag finns det ett stort antal enheter som kör Android. Dessa enheter har flera olika skärmstorlekar och upplösningar. De har olika hårdvara i stort med olika fysiska knappar på olika enheter (Andersson K., 2011). Att det finns många olika versioner av Android på alla dessa enheter gör att det blir mycket att anpassa sig till. Det här gör att vid utveckling till enheter som har Android måste noggranna tester göras innan publicering för att undvika fel. Det stora antal varianter av telefoner som idag har Android gör att det är omöjligt att testa sina appar på alla enheterna. På Android Market presenteras en lista över vilka telefoner som är de som laddat hem den utvecklade appen mest. Då användaren alltid skall

vara i fokus ger denna lista en vägledning i vilka enheter som tester borde utföras på (Andersson K., 2011).

5.2.3 Multiplattformslösningar

Utvecklingen till mobila plattformar har blivit populärt och då det finns flertal plattformar har det tagits fram verktyg som skall förenkla utveckling till just flera plattformar. Dessa lösningar är webbaserade och har endast en kodbas. Multiplattformslösningar finns i två olika typer. Den ena är semi-nativlösningar och den andra är mobila webbaserade lösningar. Dessa står bredvid nativeutveckling, som avser utveckling till en plattform i plattformspecifik miljö. En nativeutvecklad app är således direkt anpassad till den plattform den är skapad för.

Semi-nativlösningar är produkter där utveckling sker i en miljö som sedan portas till respektive plattform. De två mest populära semi-nativlösningarna är Titanium och PhoneGap. Med semi-nativlösningar fås tillgång till systemfunktioner som bland annat GPS och filhantering men funktionerna generellt ligger alltid lite efter native-utveckling. Apparna som utvecklas med semi-nativlösningar får ett utseende som är som native-appar.

Mobilt webbaserade lösningar körs helt i enhetens webbläsare. HTML5, som är den kommande standarden för webbutveckling, ger stora möjligheter för utvecklare att ta fram hemsidor som är väl anpassade till enheter som visar dem, med en så kallat responsiv design. Detta tillsammans med andra fördelar med HTML5 från föregående versioner av HTML ger stora möjligheter till utvecklare att utveckla mer dynamiskt material (HTML5 Tutorial, 2012). Dessa mobila webbaserade lösningar har inte lika stor tillgång till systemfunktioner och hårdvarukoppling och har inte heller ett användargränssnitt som är exakt som native-appar. Exempel på mobila webbaserade lösningar är jQuery Mobile och Sencha Touch.

Nativlösningar är bättre på att anpassa sig till det aktuella systemet. Användargränssnittet är något som blir svårt att anpassa till de respektive systemen med multiplattformsverktygen och därmed blir användarvänligheten inte lika plattformslik. Många användare vill att apparna skall vara iPhone-aktiga eller Android-aktiga då de använder dem (Ramberg, 2011; Johansson C. V. & Funayama, 2011). Användarvänligheten kan även bli lidande då lösningarna som är utvecklade med multiplattformsverktygen inte har samma prestanda som de hade haft om de blir utvecklade som nativelösningar istället.

Fördelar med multiplattformslösningar är att de har en kodbas till flera plattformar vilket ger en fördel under både det första utvecklingsskedet men även senare när apparna skall underhållas. Lika som med att utvecklaren förlitar sig på de olika plattformarnas teknikaliteter när man utvecklar native-appar förlitar man sig här på utvecklarna av multiplattformslösningar. Det gör att man blir beroende av ännu en aktörs villkor och förändringar. Särskilda problem kan uppstå då funktioner med hårdvarukoppling

skall utvecklas. Tanken med dem är mycket god men de är ej mogna än. Dessutom kan en multiplattformslösning generera kod som är svår att felsöka och förvalta vilket kan skapa bekymmer när en utvecklare behöver arbeta direkt mot koden. Fördelarna med multiplattformslösningar är den enda kodbasen men även att språken som används kan vara mer välkänt för utvecklare, och då specifikt mer kända än Objective-C som används för utveckling till iOS. Om appen är av enklare slag som inte kräver större hårdvarukoppling kan det finnas en fördel i att utveckla i multiplattformslösningarna men det innebär alltid en risk (Andersson F., 2011; Andersson K., 2011; Sommerstad, 2011). Det måste göras en avvägning mellan för- och nackdelar hos de olika lösningarna för att ta ett beslut. De apputvecklare som intervjuats har alla testat multiplattformslösningar men ingen har gått över till dem på grund av den osäkerhet som det innebär. Om en enklare app skall utvecklas är verktygen lämpliga men i mer avancerade appar finns en risk att multiplattformslösningen inte räcker till och då är utvecklingsprocessen tillbaka på noll.

5.2.4 Webbtjänster

Med webbtjänster menas en dynamisk interaktion mellan olika datortjänster. I fallet handlar det om att appen skall ansluta till Scantias servrar för att hämta data om de fordon som skall presenteras. Samma interaktion sker när data skall hämtas från serverna när ett åkeri loggar in på portalen och fordonen presenteras där.

Det finns flertal protokoll för detta och idag använder sig Scantias Fleet Managent-sektion av protokollet *SOAP* som bygger på standarden XML. Ett nyare protokoll som de flesta mobila enheter har stöd för heter *REST*. REST är ett protokoll som skall vara enklare att implementera och använda. Det blir även mindre data som överförs för samma information vilket leder till snabbare system och mindre belastning på dem.

Då appar skall anslutas till existerande system måste oftast några modifieringar göras. Om dessa hanteras av andra enheter leder det till att utvecklaren blir beroende av andra parter, precis som tidigare nämnt. Det är ofta i sådana situationer som projekt har dragit ut på tiden. Därmed är det av stor vikt att planera för vilka förändringar som måste göras för att sedan se till att förändringarna i serverna blir gjorda.

5.2.5 Publicering av appar

Vid önskan om att publicera sina iOS-appar måste ett Apple-utvecklarkonto skapas (Apple Developer - Tools Workflow Guide for iOS, 2012). Som enskild person finns det bara ett val av utvecklarkonto. Vid publiceringsskedet skickas den utvecklade appen till Apple som därefter måste godkänna den. Om de inte har några invändningar publiceras appen på App Store. Ett svar på godkännande från Apple tar cirka en vecka. Att vara en registrerad Apple-utvecklare som privatperson kostar 99 dollar per år. För företag finns det två olika typer av utvecklarkonton. Det ena är likt kontot beskrivet för privatperson men för företag kan ett helt team vara medlemmar under ett gemensamt utvecklarkonto. Precis som för

en privatperson kostar det 99 dollar per år för ett sådant konto (Apple Developer - Apple Developer Program Enrollment, 2012). Det andra alternativet för företag är att skapa ett Enterprise-konto. Med det kan företaget lansera appar utan att Apple kontrollerar dessa, men apparna kommer inte heller att finnas på App Store. Företaget kan istället distribuera apparna in-house till medlemmar inom organisationen i en motsvarighet till en privat App Store. Att ha ett Enterprise-utvecklarkonto kostar 299 dollar per år (Apple Developer - iOS Developer Enterprise Program, 2012). En möjlighet är att äga båda typer av konton vilket kan vara fördelaktigt då det går att testa sina appar snabbt med Enterprise-kontot och när företaget vill lansera dem gör de det via sitt vanliga utvecklarkonto (Andersson K., 2011). För att kunna publicera en app på App Store under ett företagsnamn krävs det att företaget styrker sin identitet, vilket kan bli en administrativ uppgift med smärre ledtider (Zandelin, 2011).

När en Android-app skall publiceras måste den signeras med ett certifikat där utvecklaren har den privata nyckeln (Android Developers - Preparing for Release, 2012). Appen kan därefter publiceras och installeras av användare på många olika sätt, allt från en egen webbserver till Android Market. För att publicera appen på Android Market måste ett Google-konto skapas. Kostnaden för att registreras som utvecklare är 25 dollar (Android Developers - Preparing for Release, 2012). När det är gjort kan appen laddas upp på Android Market där den även kan uppdateras när företaget så önskar (Android Developers - Preparing for Release, 2012). Det sker ingen extern granskning av appen och den läggs även ut på en gång på Android Market. Det hela gör att det går snabbt att publicera på Android Market men det finns heller inte något kvalitetskrav (Zandelin, 2011).

Tabell 3 visar en sammanfattning över de refererade intervjuobjekten i detta avsnitt.

Intervjuobjekt	Företag
Carina V. Johansson & Tina Funayama	<i>Taxi Stockholm</i>
Erica Zandelin	<i>Scania Corporate Relations</i>
Erik Ramberg & Kristoffer Sommerstad	<i>SWAD (numera Springworks)</i>
Fredrik Andersson	<i>Kondensator</i>
Kenneth Andersson	<i>The Mobile Life</i>

Tabell 3: Sammanfattning av intervjuobjekt samt företag

5.3 Val av plattform

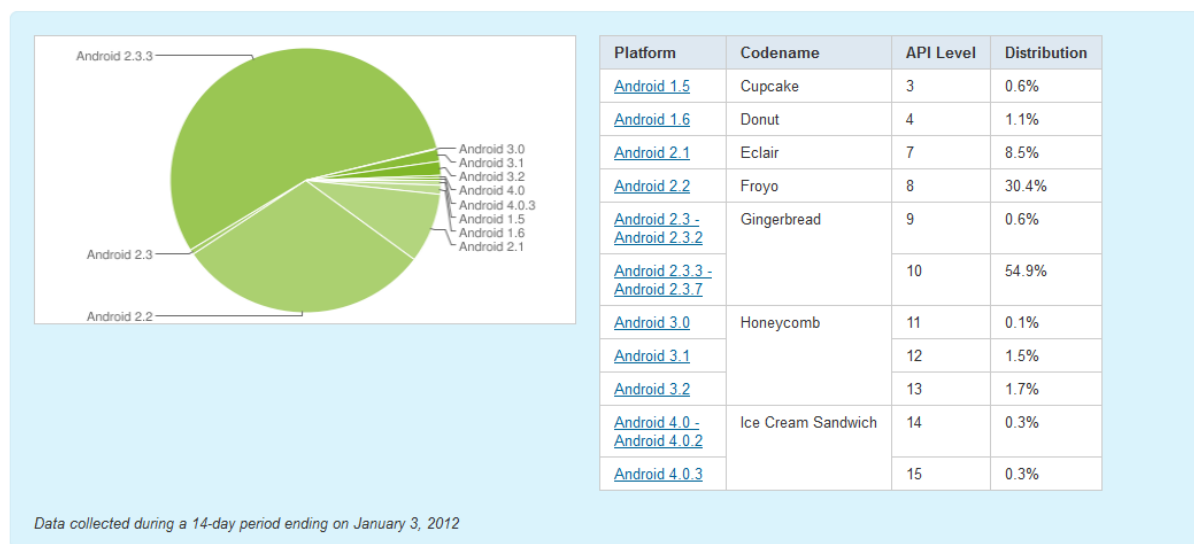
Att det finns flera plattformar att välja på ger delvis huvudbry men även möjligheter för utvecklare och användare. Något som alla de intervjuade har varit överens om är att vid utveckling måste fokus ligga på användaren. Vem som är den slutliga användaren av appen och hur den kommer att använda sig av denna är något som inte får förbises. Detta gäller även valet av mobil plattform. En fråga som därmed även blir aktuell är om utveckling skall ske till flera plattformar eller endast en. Valet av en plattform gör att appen vid publicering syns i den valda plattformens mobilbutik, även det är något som i sig påverkar vilken som blir gruppen av människor som företaget når ut till. Oavsett är det väldigt viktigt att synas i sådana butiker med sina utvecklade appar då det är ett bra sätt att exponera sig på (Slott, 2011).

5.3.1 Mobila plattformar

Idag visar statistik och det som har framkommit i intervjuer med apputvecklare att det är mobilplattformarna Android och iPhone som är marknadsledande idag och de utvecklarna bör satsa på. Plattformen Symbian förutspås försvinna (Petty & Stevens, 2011b). BlackBerry är en intressant plattform som har sin största marknad i Nordamerika (Paczowski, 2010). Då Scantias marknad där är försumbar, speciellt inom fordonsområdet, blir denna plattform ointressant. En sista intressant plattform är Windows Mobile som förutspås kunna lyckas, både av Gartner men även av intervjuobjekten inom apputvecklingsbranschen (Petty & Stevens, 2011b). Windows Phone befinner sig dock fortfarande i ett tidigt skede och är ännu bara en uppstickare som har potential och därför studeras inte utveckling till denna plattform mer ingående. För att summera beskrivs endast tillvägagångssättet för att starta upp apputveckling till plattformarna iOS och Android. En gemensam åsikt hos de intervjuade personerna har även varit att idag kommer man långt med att endast utveckla till iOS men Android har visat en sådan stark utveckling att marknaden kräver båda plattformarna. De båda plattformarna bygger på olika prismodeller för kunden där iOS-kunder är mer villiga att betala för appar (Andersson F., 2011; Resare, 2011).

5.3.2 Val av plattformsversion

Idag kommer det ofta nya versioner av Android vilket gör att användarna finns utspridda över dessa. Det här gör valet av plattformsversion viktigt. Med de högre versionerna av plattformar erhålls större funktionalitet men användarskaran blir mindre och dessutom existerar begränsningar i vilka telefoner som går att uppdatera eller inte. Därför måste en avvägning göras angående vilken plattformsversion utvecklingen bör ske till. Fördelningen för de nuvarande plattformsversionerna hos Android finns presenterade i figur 18. För att summera rekommenderas att utveckla mot version 2.1 av Android om alla önskade funktioner stöds i den versionen. Om så inte är fallet får utvecklingen ske till den lägsta version där de önskade funktionerna fungerar.



Figur 18: Illustration över fördelningen bland de nuvarande Android-versionerna (Android Developers - Platform Versions, 2012)

Idag använder de flesta sig av version 2.1 eller högre (Android Developers - Platform Versions, 2012). Det motiverar till att utvecklingen borde ske till just denna version för att göra appen tillgänglig för så många användare som möjligt. Detta är även något som Android själv rekommenderar för att nå ut till ett så stort antal användare som möjligt. Om en viss funktionalitet inte finns tillgänglig i den versionen måste en avvägning göras mellan funktionalitet och användartillgänglighet.

För iOS är det svårare att hitta tydlig statistik över andelar av versioner och därmed även svårare att rekommendera en version att utveckla till. Rekommendationen är att alltid använda sig av den senaste versionen för att få tillgång till de nyaste funktionerna i utvecklingen. Vet utvecklaren att många användare inte har uppdaterat till de senaste versionerna är det rekommenderat att se över om de funktioner som den utvecklade appen använder sig av stöds i de äldre versionerna.

Tabell 4 visar en sammanfattning över de refererade intervjuobjekten i detta avsnitt.

Intervjuobjekt	Företag
Fredrik Andersson	<i>Kondensator</i>
Ingemar Resare	<i>Sveriges Åkeriföretag</i>

Tabell 4: Sammanfattning av intervjuobjekt samt företag

6 Fleet Management-prototypen

I detta kapitel presenteras empiri som har inhämtats via de trafikledare som har intervjuats på åkerierna. Kapitlet inleder med hur Fleet Management-webbportalen används idag och övergår därefter till vilka funktioner som skulle vara intressanta i en mobil variant. Slutligen beskrivs den prototyp som har utvecklats till iOS och responsen från åkerier har tagits i beaktande under prototyputvecklingen.

En diskussion kring den erhållna behovsbilden hos Fleet Managements slutanvändare är intressant då vi även testat principen att beakta den potentiella slutanvändarens behov under utvecklingen. En poäng som har framkommit ett flertal gånger i det förra kapitlet är slutanvändarens viktiga roll vid apputvecklingen och framtagningen av en Fleet Management-prototyp bygger således på denna observation.

6.1 Användandet av FMP-portalen

Hos åkerier används idag Fleet Management-portalen primärt av två olika roller. Dessa är åkeriägare och trafikledare (Stjärnström, 2011). Åkeriägare är mer intresserade av uppföljning av fordonen: de vill veta hur fordonen mår och använder sig därmed av de rapporter som finns tillgängliga på portalen. Att hålla nere servicekostnaderna på fordonen är därmed ett prioriterat område.

Trafikledare är intresserade av fordonens position. I gruppen av trafikledare finns det två olika typer beroende på åkeriets verksamhet. Det första är åkeriet som kör linjetrafik, där endast samma sträcka körs, exempelvis varje dag. Trafikledare på sådana åkerier behöver inte ständigt följa fordonen då de vet var fordonen borde vara. För sådana trafikledare är positionering nyttigt om någonting speciellt har hänt, till exempel en bil inte är där den borde vara (Jansson, 2011), eller i situationer där trafikledaren vill få en snabb överblick av flottan för att bekräfta att allt är som det skall (Bogeling, 2011; Sunding, 2011), alltså ett mer rutinmässigt användande. Den andra typen av åkerier är de som kör ordertrafik. De använder positioneringen i ett planeringssyfte där de fortlöpande får in ordrar från kunder och dessa skall placeras ut på fordonen som finns på vägarna.

Om situationen är sådan att åkerier har trafikledare som ständigt sitter vid en dator med webbportalen blir mobiltelefonen överflödig. För en person som däremot har flera roller på åkeriet, säg trafikledare, åkeriägare samt serviceansvarig, och därmed inte sitter vid datorn hela tiden ger appen en flexibilitet och mobil tillgänglighet av Fleet Management-tjänsterna. Vissa trafikledare på de intervjuade åkerierna har testat att använda sig av den existerande webbportalen i sin mobiltelefon i ett vanligt webbläsarfönster, men de anser att det har varit plottrigt, att det tar för lång tid, det är smått och inte alls användarvänligt eller inte alls fungerat (Jansson, 2011; Sunding, 2011).

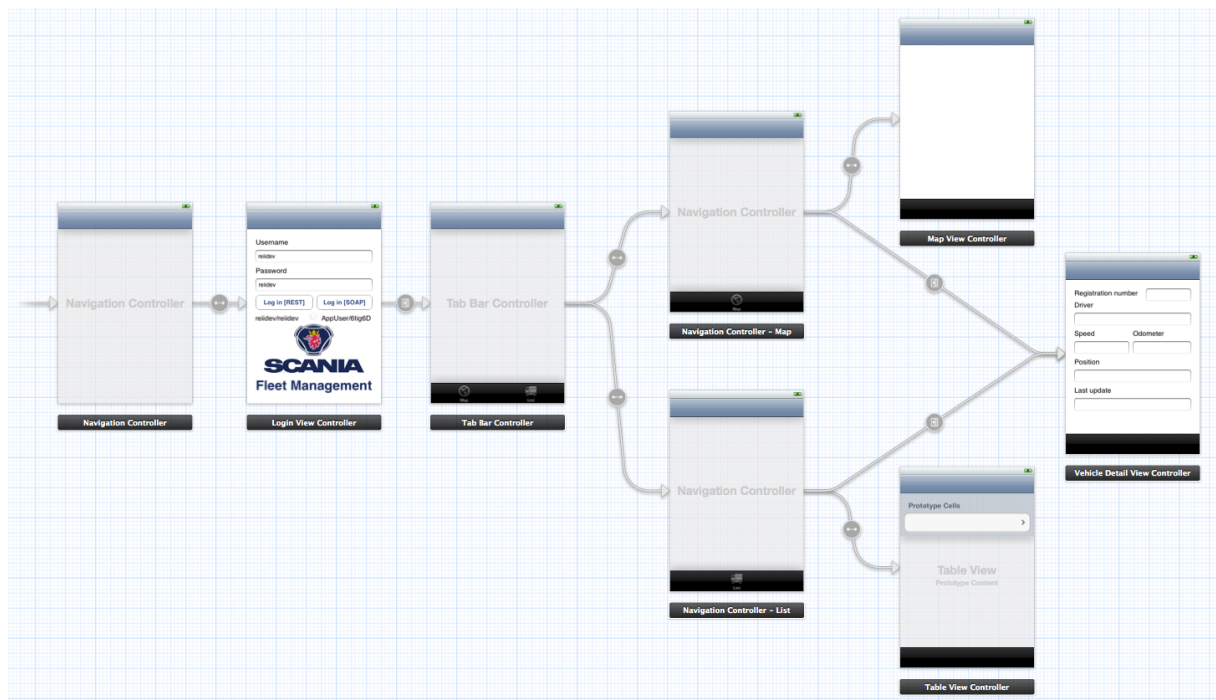
6.2 FM-appens potentiella funktionalitet

En gemensam respons från alla de intervjuade åkerierna har varit att positioneringen av fordonen är den viktigaste funktionen. Andra data som de anser är bra att visa är hastighet, mätarställning och kvarvarande bränsle i tanken.

Vid flera tillfällen har diskussioner uppstått kring att en app för chaufförerna vore intressant. Scania har idag en interaktiv produkt, en Interaktorn, som finns installerat i vissa äldre fordon. Till den kan trafikledaren sända orderinformation och i övrigt kommunicera med föraren. Interaktorn håller idag på fasas ut och en app skulle vara en bra ersättningsenhet som uppfyller samma syfte. Om en Interaktor, eller ett motsvarande system, idag inte finns installerat i bilen sker kommunikationen angående ordrar via telefon eller SMS, vilket blir en kostnad för åkerierna (Stjärnström, 2011; Bogeling, 2011). Med tanke på att många åkerier kör internationell trafik kan dessa kostnader bli höga. Det bör dock noteras att i en app för chaufförerna kommer datatrafik att sändas till mobiltelefonerna vilket i sig kan bli en hög roamingkostnad utomlands. Utöver orderhantering har det diskuterats att chaufförerna skall kunna se en poängsättning på sig själva utifrån hur bra de kör med tanke på skonsamhet mot fordonet och miljön, den tidigare presenterade trafikljusrapporten. Det har visat sig att de chaufförer som har fått ta del av denna information har därefter själva visat intresse för om de har förbättrat sig. Det har både handlat om att förarna vill förbättra sitt eget körsätt men det har även uppstått ett tävlingsmoment och yrkes stolthet där chaufförerna vill få bättre resultat än sina kollegor (Bogeling, 2011; Åhlund, 2011). Andra fördelar en app har jämfört med Interaktorn är att smartphones är noterbart billigare enheter än Interaktorn och att en telefon just är mobil. En chaufför kan ta med sig den utanför bilen och behöver därmed ej sitta i bilen och vänta på ordrar utan kan istället besöka ett fik eller liknande (Bogeling, 2011). Det skulle leda till bättre arbetsmiljö för chauffören.

6.3 Utveckling av en Fleet Management-prototyp

Prototypen i denna studie är skapad i Xcode till iPhone och iPad. Med Xcode finns möjlighet att skapa en app för iPhone och iPad med en gemensam kod. Dock behövs layouten på appen göras två gånger på grund av de olika skärmstorlekarna på enheterna. Layouten byggs i en funktion som kallas *storyboard*. En storyboard visar ett flöde av vyer som gör det enkelt för utvecklaren att överblicka de olika vyernas koppling till varandra. Figur 19 visar storyboarden för iPhone-versionen. I iPad-versionen finns endast ändringar i placering av komponenter på en synlig yta för användaren.

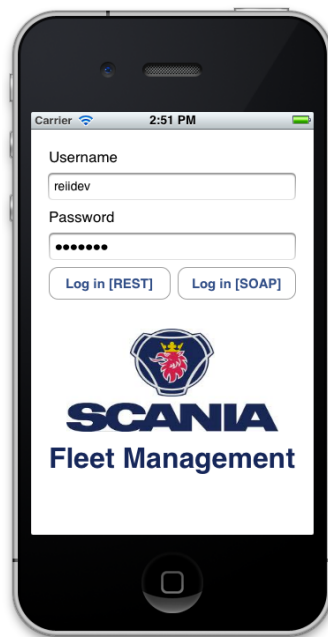


Figur 19: iPhone-storyboard för Fleet Management-prototypen

I storyboarden finns vissa kontroller som inte blir explicit synliga för användaren men som appen använder sig av: i Fleet Management-appen används komponenterna Navigation Controller, Tab Bar Controller och Table View. Navigation Controllern hanterar de olika flödena i appen och Tab Bar Controllern ger utvecklaren en möjlighet att implementera en vy med flera tabbar. I en Table View infogas celler som sedan blir en modell för hur tabellen skall presentera cellen för användaren. De övriga vyerna som är synliga för användaren presenteras var för sig med upplägg och funktionalitet.

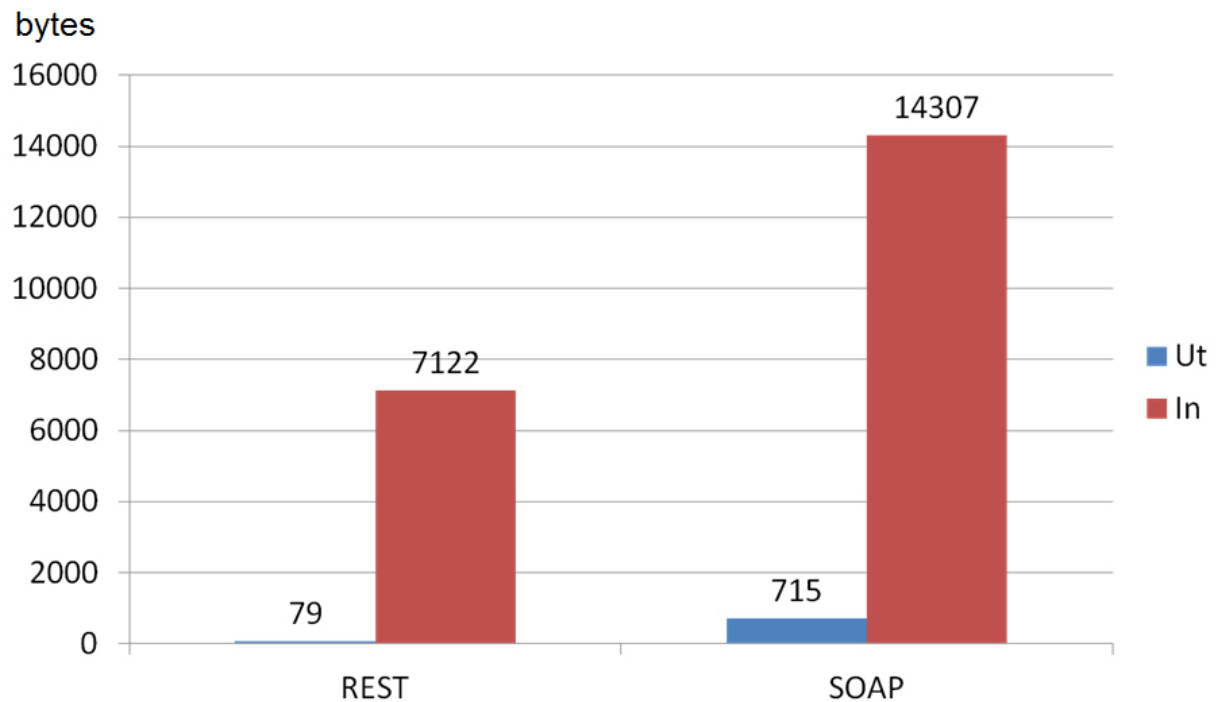
Loginvyn

Den inledande vyn vid appens uppstart är loginvyn. I den ombeds användaren att mata in ett användarnamn och ett lösenord. Figur 20 visar vyn som användaren möts av vid uppstart av appen. Då appen är en prototyp, och ett syfte med appen var att testa att ansluta sig med olika webbtjänster mot olika typer av servrar, finns det två loginfunktioner som just använder sig av dessa två protokoll.



Figur 20: Loginvyn för Fleet Management-prototypen

De olika webbtjänsterna är, som tidigare har nämnts, SOAP som Scania's Fleet Management-sektion idag använder sig av och REST som är en nyare typ av webbtjänst och som de olika mobila plattformarna har bättre inbyggt stöd för. En REST-server finns uppsatt i testsyfte för att testa anslutning med REST-protokollet. Då det handlar om två olika protokoll kan de även påverka i användningsfallet. Något som är en tydlig skillnad är datapaketens storlek som sänds och tas emot vid autentisering och mottagning av fordonens information. I prototypen sker autentisering genom att användarnamn och lösenord skickas och om dessa är rätt skickar servern tillbaka information och alla åkeriets fordon. Figur 21 visar resultatet på paketens storlek för denna trafik. Det går att urskilja att paketens storlek vid användandet av SOAP blir större än vid REST. Dock är paketen så små att användarens väntetid inte påverkas avsevärt. Medan denna sändning sker finns även en aktivitetsindikator utplacerad för att göra användaren medveten om att appen arbetar. Denna indikator blir synlig och roterar endast då en inloggningsknapp trycks på och försvinner om loginvyn försvinner eller om ett felmeddelande dyker upp. Felmeddelandet kan handla om fel inloggningsuppgifter alternativt anslutningsproblem.

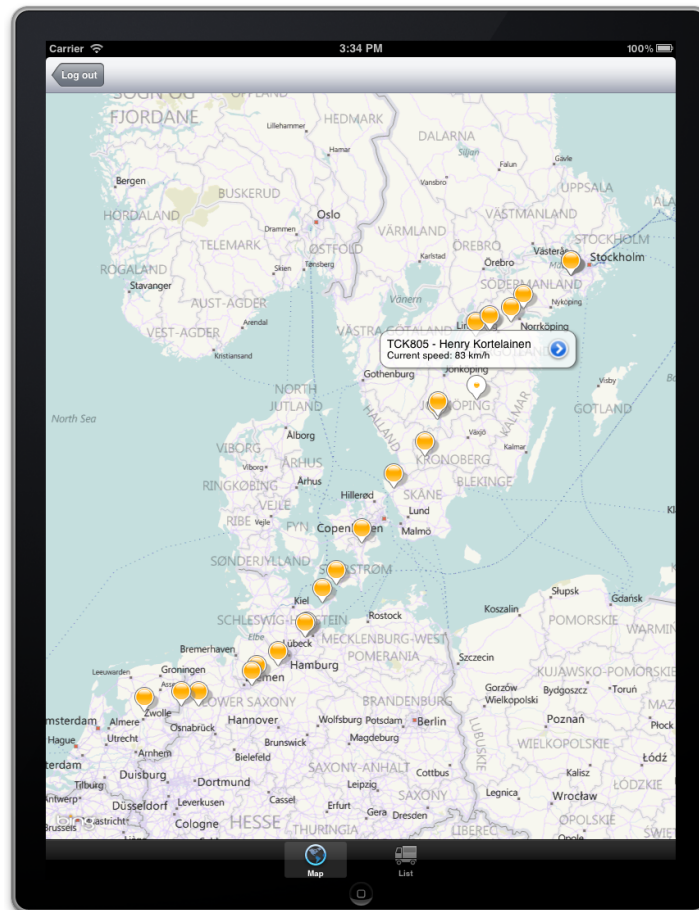


Figur 21: Jämförelse av paketstorlek då REST respektive SOAP används

Vid rätt inmatning av användarnamn och lösenord verifierar servern inloggningsuppgifterna och om dessa blir godkända skickar servern tillbaka information om alla de fordon som finns i flottan. Dessa data lagras i appens delegat, detta på grund av att delegatet är en central komponent i appen (Apple Developer – Cocoa Core Competencies: Delegation, 2011) och därmed kan alla delar av appen lätt komma åt delegatet och det data som är lagrat där. Vyer som hämtar data om alla fordonens information från delegatet är kartvyn och listvyn.

Kartvyn

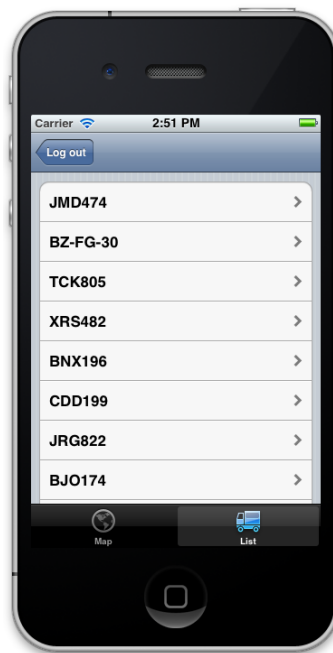
Kartvyn är skapad med hjälp av Bing Maps som är en kartlösning av Microsoft. Valet av Bing Maps före det mer storskaligt använda Google Maps beror på att den nuvarande portalen för Fleet Management använder sig av Bing Maps. För att kunna implementera Bing Maps i egna iOS-appar måste Bing Maps SDK laddas hem och läggas till i projektet. Varje nål på kartan representerar ett fordon och data till dessa nålar hämtas från appens delegat som tidigare lagrats vid inloggning. Dessa nålar är klickbara och om en nål klickas på skapas en bubbla med info om det fordonet. Bubblan visar registreringsnummer, förare och aktuell hastighet. Om pilen i bubblans högra kant klickas visas en detaljvy som visar mer detaljerad info om fordonen. Kartvyn visas i figur 22. Kartan är en flik av två tillgängliga där den andra är en listvy. För kart- och listvyn finns alltid en knapp i övre menyn där användaren kan logga ut om så önskas.



Figur 22: Kartvyn för Fleet Management-prototypen

Listvyn

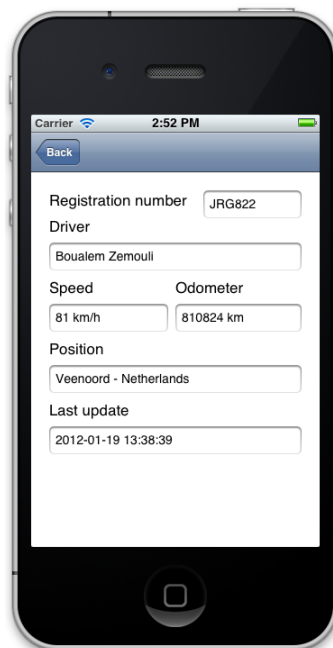
Precis som kartvyn hämtar listvyn sina data om fordonen från appens delegat. Listvyn presenterar endast registreringsnummer på fordonen, precis som figur 23 visar. Varje rad i listan är klickbar och om en rad väljs av användaren skapas en ny detaljvy med mer detaljerad info om fordonet. Detta är samma detaljerade vy som tas fram om en användare väljer att visa mer info om ett fordon via knappen i bubblan för varje bil.



Figur 23: Listvyn för Fleet Management-prototypen

Detaljvyn

När detaljvyn skall visas kan den hämta information om det valda fordonet från appens delegat och därmed presentera detta. I detaljvyn visas även en bakåtknapp som tar användaren tillbaka till vyn som visar kartan eller listan, beroende på var användaren kom ifrån. Figur 24 visar ett exempel på detaljvyn för iPhone.



Figur 24: Detaljvyn för Fleet Management-prototypen

Testning

Det är möjligt att antingen testa en utvecklad app i en simulator på datorn som utvecklingen sker på eller på en enhet. För att testa appen i en simulator finns det inga krav men om en testning skall göras på enheter måste ett Apple Developer-konto införskaffas och enheten registreras som en test-enhet. För prototypen utfördes endast manuella testar då den enkla funktionaliteten tillåter detta. Vid utvecklingen av denna prototyp har det existerande Scania-kontot för App Store använts och som användes för publicering av Scania Dealer Locator-appen.

6.4 Återkoppling från användarna

Ett besök gjordes hos Joakim Jansson på Scania Transportlaboratorium för att få återkoppling till prototypappen. De generella åsikterna var positiva. Han ansåg att appen var behändig, snabb och användbar. Angående kartvyn var responsen att markörerna som visar en bils position även skulle visa riktningen bilen är på väg mot och en annorlunda markör om en bil är igång eller inte skulle visas. Jansson (2012) önskade även att tiden då bilens information senast uppdaterades kunde ses direkt i bubblan på kartan istället för i detaljvyn. Dessa funktioner är möjliga att implementera med det data som fås ut idag vid datahämtning om fordonen. En ytterligare funktion kunde vara att användaren kan se bubbelinformation om alla fordonen samtidigt för att få en snabb överblick över var alla fordonen befinner sig, samma info som annars fås fram för att på ett fordon genom att klicka på en markör. I listvyn önskade Jansson (2012) att han kunde söka på registreringsnummer och förare.

Tabell 5 visar en sammanfattning över de refererade intervjuobjekten i detta kapitel.

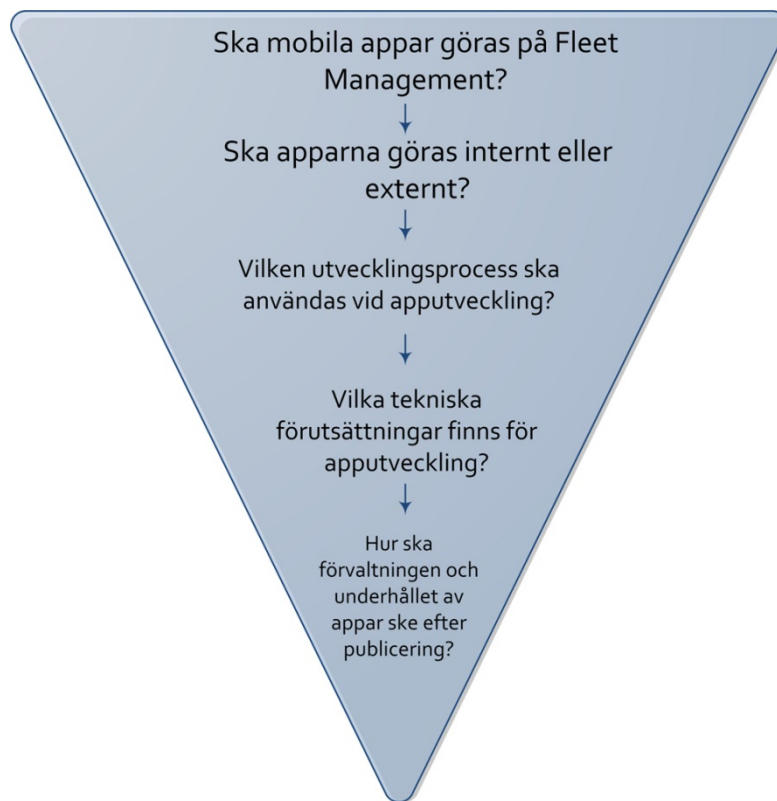
Intervjuobjekt	Driftprovskund
Frida Stjärnström	<i>Stjärnströms International AB</i>
Joakim Jansson	<i>Scania Transportlaboratorium AB</i>
Jeff Thomas Bogeling	<i>LTD Spedition AB</i>
Curt Sunding	<i>CeEss Transport AB</i>
Roger Åhlund	<i>Scania-Bilar Sverige AB</i>

Tabell 5: Sammanfattning av intervjuobjekt samt driftprovskunder

7 Analys

I detta kapitel presenteras den genomförda analysen som sammanväver den empiriska undersökningen med den teoretiska referensram som användes för studien.

Den analysmodell som används i denna studie bygger på att svara på de olika frågorna enligt ett sekventiellt tillvägagångssätt. Analysen bygger på en trattmodell som bryter ned frågeställningarna till mer konkreta, lättbegripliga och lätthanterliga diskussionspunkter som rör de förutsättningar och utmaningar som finns ur både ett processperspektiv och ett tekniskt perspektiv. Se figur 25 för en visualisering av analysmodellen.



Figur 25: Illustration över den analysmodell som har använts för studien

7.1 Att utveckla mobila appar

Studiens stora fråga är om Fleet Management-sektionen skall utveckla appar till mobila enheter, precis som de utvecklar en webbportal idag. Svaret på denna fråga har en avgörande roll för resten av studien och är därmed även av största vikt.

För en lyckad app måste det finnas en behovsbild. En Fleet Management-app har visat sig ha störst behov hos individer som har flera roller på ett åkeri där en av rollerna är trafikledare. Det kan till exempel handla om ett åkeri där personen sköter trafikledning, underhåll av fordon samt har en chaufförsroll. Individens arbete på åkeriet skall inte enbart bestå av att arbeta vid en dator vilket då skulle göra att information om fordonen lika väl kan ses på datorskärmen som på en mobiltelefon. Som tidigare har presenterats är en stor del av Scantias kunder åkerier där antalet fordon är av sådant antal att appen skulle komma till användning. Det här utesluter inte möjligheterna för en trafikledare att på heltid ha användning av en mobil app då tillgängligheten till fordonsinformationen blir väldigt enkel på en mobil enhet. Oavsett i vilken bransch en organisation verkar i och som överväger att utveckla kompletterande appar för existerande tjänster bör dessa också analysera sina potentiella slutanvändares behovsbild och se hur appen skulle kunna komma tillrätta.

Vid beslut om att utveckla en app för Fleet Management måste beslut tas om appens omfattning angående funktionalitet. För att få en klarhet i denna fråga är det viktigt att integrera de verkliga slutanvändarna av appen. Under studiens gång har diskussioner med Scantias driftprovskunder skett angående appens funktionalitet. Sektionen för Fleet Management har en utmärkt möjlighet i att fortsätta diskutera denna fråga med sina driftprovskunder, både när inledande funktionalitet skall bestämmas men även efter en lansering för återkoppling. Rekommenderat är att de prototypappar som tidigare tagits fram ligger som grund för diskussionerna kring appens funktionalitet. Det är slutanvändarnas åsikter som är de viktigaste och måste alltid prioriteras före funktionalitet. Appens första version behöver inte heller innehålla all den funktionalitet som beslutats för appen utan kan implementeras i senare versioner. Detta görs då även responsen från användarna kan påverka vilka funktioner som borde implementeras i nästkommande versioner. Vid en iterativ process med slutanvändaren försäkras utvecklaren sig mot att överflödig funktionalitet utvecklas. Detta går hand i hand med en av Scantias kärnvärderingar som är eliminering av slöseri och som utgör basen i R&D Factory-modellen.

Det är även av vikt att förankra appen internt. Förankringen skapar en gemensam förståelse och skall vara på en nivå som inte skapar en omöjlig situation med många åsikter. Förankringen skall inte heller vara för liten då initiativtagaren i ett sådant läge kan tappa möjligheten att få med appen i ett större perspektiv och då planering sker. Det är viktigt att ha en långsiktig plan. Efter den första publiceringen av appen kan inte appen läggas bakom sig utan tekniska förändringar hos leverantörerna av

mobiltelefoner och plattformarna som finns på mobiltelefonerna gör att apparna måste underhållas efter publicering. En fortsatt diskussion kring appens förvaltning och underhåll sker senare.

I dagsläget finns det en app som är lanserad av Scania till iPhone och Android, Scania Dealer Locator. Många konkurrenter har redan lanserat appar till sina respektive Fleet Management-lösningar. Det går inte komma ifrån att appar är en trend som många har anammat. Då en app skulle öka Scantias tjänsteutbud bidrar det till en ytterligare konkurrerande faktor i branschen. Eftersom många av konkurrenterna idag redan har en app kan kunden anse det vara en självklarhet med en kompletterande app för att övervaka flottan. Att publicera en app tillför en publicitet och visar att företaget är med i teknikens framkant. Publiciteten blir extern mot marknaden som visar att Scania följer tekniska trender men även en intern publicitet skapas. Då Scania är en stor organisation blir det naturligt att avdelningar får anstränga sig för att synas på de högre nivåerna inom organisationen, detta på grund av att prioriteringar alltid måste göras angående vilka tekniker som ska satsas på framöver. Då sektionen för Fleet Management är en sektion som till stor del utvecklar mjukvara finns det här en stor möjlighet att visa framfötterna på ett sätt som uttrycker ambition på att anamma de tekniker som går att anpassas till verksamheten. I Scantias fall knyter det även an till teori som berör tjänster och tjänsteutbud: det blir viktigt att hitta extratjänster som kompletterar de fordon som säljs och som förväntas ha långa livslängder. En app har sällan en kannibaliserande inverkan på övriga tjänster, och agerar främst som ett komplement som kan komma användaren till nytta.

7.2 Intern eller extern utveckling av mobila appar

En annan stor fråga som har berörts i denna studie är om appen bör göras internt eller om en extern leverantör bör ta ansvaret för utvecklingen av appen.

I den empiriska undersökningen framkom det att alla benchmark-företag valde att lägga ut apputvecklingen hos en tredjepart med den rätta kompetensen. Hur mycket IT-verksamhet som bedrevs internt hos företagen varierade: vissa företag valde att outsourca hela sin IT-verksamhet medan andra företag bedrev mycket kritisk utveckling internt. Med kritisk utveckling menas den utveckling som direkt knyter an till ens kärnkompetens och huvudsakliga fokus och som bedrivs in-house för att behålla kontrollen.

Trots att mycket IT-relaterad utveckling har skett internt hos vissa av företagen valde de ändå att outsourca utvecklingen. De under intervjuerna erhålla resonemangen knyter direkt till den teori som berör outsourcing av mjukvaruutveckling, och som Quinn (1999; 2000) presenterar. Externa apputvecklare agerar således en snabb källa till kunskap, och kundens osäkerhet minskar om apputvecklaren har den spetskompetens som eftersökes och en portfölj som visar prov på att de klarar av att producera appar som skall användas i ett verksamhetsnära syfte.

Baserat på de svar som har erhållits under studiens lopp samt den teori som berör ämnesområdet kan det vara en god idé att agera via en tredjepart inledningsvis för att erhålla all den kompetens och kunskap som de kan erbjuda, speciellt när det handlar om att utforma heltäckande workshops så att behovsbilden klargörs. Mobil spetskompetens är svår att finna då utveckling till de populäraste mobilplattformarna idag förutsätter spetskompetens inom nischade programmeringsspråk såsom Objective-C, och den kompetens som existerar på marknaden brukar självmant söka sig till mindre avenyer där deras spets kan utnyttjas på bästa sätt. Således kommer de mest kompetenta apputvecklarna att söka sig naturligt till apputvecklingsföretag där deras spetskompetens kan komma tillrätta. Att finna egen spetskompetens är både dyrt och svårt, särskilt eftersom en nyanställd måste fortbildas och sättas in i det rådande arbetssystemet. Om appen ej skulle få det genomslag som har förväntats, det vill säga bli en viktig del i organisationens tjänsteutbud, kan det bli svårt att sysselsätta den anställda tillräckligt mycket, särskilt om appen behöver förhålla sig till ett annat system som exempelvis Fleet Management-portalen. Dessutom måste flera typer av kompetens erhållas, exempelvis en Objective-C-utvecklare, en utvecklare för Android-plattformen, en grafiker och en specialist på mobil interaktionsdesign. Att finna all denna kompetens blir snabbt kostsamt och mödosamt. Nackdelen är dock att företaget går miste om den kunskap som hade erhållits om utvecklingen skedde internt, till exempel goda insikter i hur interaktionsdesignen skall se ut eller hur man på bästa sätt tar till sig slutanvändarens synpunkter vid formgivningen av en app, kunskap och kreativitet som skulle kunna införlivas i andra aspekter av det arbete som genomförs på avdelningen.

På Scania Fleet Management har det visat sig att förståelsen för appen generellt sett är hög, särskilt då sektionens kärnkompetens kretsar kring mjukvaruutveckling. Den problematik som Abbas et al. (1997) belyser, att en okunskap om ämnesområdet gör att den okunnige hamnar i underläge vid förhandlingar, bör således ej existera om ett beslut att outsourca apputvecklingen sker. Om apputvecklingsförfarandet skall fasas in vid ett senare skede, till exempel för att återerövra kontrollen över något som har blivit en viktig affärsfaktor, måste flera aspekter beaktas. De direkta fördelarna av en extern parts medverkan i början av processen bör ställas mot det extra arbete det tar att fasa över arbetet till Scania vid ett senare skede. Integrationsproblematiken har varit något som har dykt upp frekvent under intervjuerna och är ett problem som typiskt underskattas. Rättigheterna till koden är också en faktor som bör beaktas, då ett övertagande av utvecklingsprocessen förutsätter att organisationen har rätt att modifiera koden.

Apputvecklare har en bra omvärldssyn och har en god förståelse för hur saker bör göras och till vilka plattformar och är ständigt uppdaterade på de förändringar som uppkommer inom området. En organisation vars kärnkompetens berör ett annat område kan ha svårt att förhålla sig till de utvecklingar som sker inom smartphone-området. Det har också visat sig att alla apputvecklare jobbar i agila utvecklingsmiljöer, vilket är en stor fördel då iterationer och feedback från slutanvändare är en förutsättning för att skapa en användarvänlig app och för att beakta förändringar som uppkommer.

Feedback behövs för att säkerställa att kunden får det den vill ha och att alla potentiella problem undanröjes tidigt. Dessutom bygger appar på interaktion i stor utsträckning, och för att säkerställa att användbarheten blir hög måste kontinuerlig användningsfeedback erhållas under utvecklingen.

Sammanfattningsvis är det viktigt ta in den rätta kunskapen på ett eller annat sätt, bland annat kunskap som rör gränssnitt, interaktionsdesign i appar samt de nödvändiga programmeringskunskaperna. Detta kan ske antingen genom rekrytering, intern utbildning eller genom att ta in en tredjepart. Oavsett vilken väg som tas är det viktigt att det inledande planerings- och analysarbetet görs på ett korrekt sätt så att behovsbilden och appens syfte är tydligt. Att behöva göra förändringar i ett senare skede av projektet är kostsamt. Det är också viktigt att förhålla sig till de snabba tekniska förändringarna som sker på smartphone-området, vilket tvingar utvecklaren att agera proaktivt när en ny uppdatering har släppts, som ett exempel. Om utvecklingen skall ske internt måste alltså resurser för att följa upp denna utveckling finnas. Om appen visar sig bli en viktig tjänst för kunden rekommenderar både teorin och intervjuade benchmark-objekt att utvecklingen och förvaltningsarbetet bör ske in-house för att försäkra kvaliteten och för att bibehålla kontrollen.

7.3 Utvecklingsprocessen

Detta avsnitt diskuterar den utvecklingsprocess som i dagsläget används på Scania, RUP, samt den utvecklingsprocess som används av nästintill samtliga jämförbara företag och apputvecklare, Scrum.

Scrum är en starkt agil utvecklingsmodell som förutsätter att utvecklingen av mjukvara sker i en oförutsägbar och föränderlig miljö. Denna beskrivning har visat sig stämmer väl överens med den rådande smartphone-miljön, där förändringar sker snabbt och på ett svårförutsägbart sätt. Teorin som kretsar kring agil mjukvaruutveckling och Scrum betonar att kreativiteten främjas om utvecklarna får prova sig fram och experimentera istället för att arbeta efter normativa processer och fördefinierade krav.

I den empiriska undersökningen har det framkommit att apputveckling inte skiljer sig så mycket från klassisk mjukvaruutveckling, utan den största skillnaden ligger i hur slutanvändaren måste ses som en viktig del genom hela utvecklingsprocessen. Det är ytterst viktigt att få med slutanvändarfeedback i alla led: kundens behov skall alltid komma först vilket kan kopplas ihop med de värderingar Poppendieck (2006) förespråkar med mjukvaru-lean. Om en specifik funktion i appen ej bringar ett ekonomiskt värde, det vill säga behovet för den funktionen finns ej hos slutanvändaren, skall den inte utvecklas. Detta tankesätt passar bra ihop med apputveckling, då de mest framgångsrika apparna ofta är de enklaste att använda. Att eliminera slöseri är också en av de grundläggande kärnvärderingarna på Scania, och lean mjukvaruutveckling förespråkas av Scantias nuvarande utvecklingsmodell. Den prototyp som togs fram som ett led i arbetet utgick från denna filosofi då de potentiella slutanvändarnas synpunkter styrde utformningen. Önskad funktionalitet utvecklades således ej.

Den spetskompetens som är nödvändig för utvecklingen av en app är bland annat utvecklare som är insatta i de nya programmeringsspråken, samt att ett mycket större fokus läggs på det grafiska och interaktionen mellan användaren och hårdvaran. Dock behövs inga stora projektteam för att genomföra ett apputvecklingsprojekt, något som intervjuresultaten bestyrker och som Schwabers (1995) Scrum-modell förespråkar. På Scania Fleet Management finns idag kravhanterare och användbarhetsarkitekter som har kunskap om utformningen av webbtjänsten, men inga stadgade roller som direkt knyter an till apputveckling finns representerade.

Scanias nuvarande utvecklingsmodell som bygger på RUP, SD-processen, har uppfattats som tungrodd och byråkratisk (Zandelin, 2011), vilket ej är förenligt med agila värderingar. Ur ett teoretiskt perspektiv är dock Scanias implementering av RUP bra, då den säger att mindre projekt endast skall ta ett urval av de artefakter som presenteras. Detta styrks även av Gibbs (2006) och Hirsch (2002) som menar att en avskalad RUP-implementering kan användas i mindre projekt med framgång. Dock förutsätts att avskalningen av RUP faktiskt sker, då teorin uppvisar en tendens bland RUP-användare att ändå behålla stora delar av ramverket. Andelen obligatoriska artefakter och roller i SD-processen är däremot förhållandevis låg och dessa är lämpliga även i apputvecklingssammanhang då apputveckling ej skiljer sig från vanlig mjukvaruutveckling på en övergripande nivå. Ur ett tidsperspektiv passar Scanias nuvarande releasecykel, fyra större släpp per år, med de exempel som har framförts under intervjuerna. Att släppa en uppdatering till en app varje kvartal ses ej som ett problem, och detta skulle även låta apputvecklingen ligga i linje med utvecklingen av Fleet Managements övriga delar. Appen förväntas vara en reflektion, eller förlängning, av den existerande webbtjänsten och skall därmed anpassa sig efter den.

Rollerna i Scanias SD-process presenteras på ett så generellt vis att apputvecklingsroller ej utesluts i Scanias mjukvaruutveckling. Dock förutsätter apputveckling att slutanvändaren involveras i högre grad, vilket är REIX-gruppens och funktionsspecialisternas samt kravhanterarnas ansvarsområde idag. På Scania Fleet Management måste de således se till att föra dialog med slutanvändaren i en större utsträckning än vad som sker i dagsläget. Appar är ett interaktivt medium och förutsätter bland annat att grafiker, interaktionsdesigners samt utvecklare arbetar nära med varandra och på ett iterativt sätt. I kontakten med kunden bör således alla tre grupper delta, då deras arbete med att ta fram en app i hög grad beror av varandra. På Scania jobbar dock funktionsspecialister och kravhanterare tätt, vilket även utvecklare och kravhanterare gör, som därmed gör att förståelsen för att iterera på detta sätt finns.

Kruchtens (2000) RUP-ramverk förespråkar också att en projektmedlem tar på sig flera roller i ett projekt, medan intervjuobjekten menar att bäst resultat erhålls då medlemmarna får ha specialistroller. Detta är även förenligt med Poppendiecks (2006) lean-teori som menar att slöseri och kvalitetsförsämring uppstår då en individ ständigt behöver byta roll inom teamet. Dock är detta ej ett problem på Scania Fleet Management, då de i dagsläget har fasta arbetsroller. Dessutom har detta

observerats bland de intervjuade apputvecklarna. Detta är något företag kan ha i åtanke om valet blir att inleda apputveckling in-house, eftersom man då, enligt den empiriska undersökningen och teorin, bör finna specialister för varje del i apputvecklingsprojektet. Ett apputvecklingsprojekt består dock av ett litet team vilket gör att visst rollskifte ändå uppstår på ett naturligt sätt.

Sammanfattningsvis innebär ett större fokus på slutanvändaren att apputvecklingsprocessen kräver ett något mer ostrukturerat och experimentellt tillvägagångssätt och där fler roller i utvecklingsprocessen bör involveras i kontakten med slutanvändaren. Scantias RUP-ramverk behöver ej modifieras i större utsträckning för att passa apputveckling, men det är kanske önskvärt att använda en lättviktig agil metod såsom Scrum för att tillåta den flexibilitet som krävs vid apputveckling, särskilt då den nuvarande modellen har uppfattats som tungrodd. Scrum behandlar utvecklingsprocessen som en svart låda, där feedback och åsikter från intressenter och speciellt slutanvändarna ständigt måste övervägas för att slutresultatet skall bli bra. Flexibilitet krävs också i en starkt föränderlig miljö såsom smartphone-marknaden, just eftersom tekniken förändras snabbt.

7.4 Tekniska val

I detta avsnitt diskuteras valet av operativsystem som apputvecklingen borde rikta sig mot. Även utvecklingstekniker angående native-utveckling kontra multiplattformslösningar diskuteras.

Val av plattformar

De tre största mjukvaruplattformar för smartphones idag är iOS, Android och RIM. Då RIM har sin största marknad i Nordamerika, där Scania har en försumbar fordonsmarknad, kan apputveckling som riktar sig mot RIM bortses. Utveckling till både iOS och Android borde ske då dessa plattformar idag är starka aktörer och spås att förbli det. De geografiska områden där Scania har sin största marknad idag har majoriteten av mobila enheterna operativsystemen iOS eller Android. Den starka tilltro till Windows Phones tillväxt gör att ögonen måste hållas öppna för den plattformens tillväxt. Inom en snar framtid kan marknaden kräva en app även till Windows Phone. Då utvecklingen bör ske till flera plattformar övergår vi till en diskussion kring hur detta skall lösas.

Vid valet av att utveckla med multiplattformslösningar kan tekniska begränsningar finnas. Detta sker inte om valet att utveckla native-appar tas. Utöver att låsa sig till endast iOS och Androids restriktioner och förändringar förbinder utvecklaren sig ytterligare till multiplattformarnas villkor. Efterlevnaden av apparna måste i sådant fall även ha ett vakande öga på förändringarna hos dessa lösningar.

Långsiktig plan

Som tidigare nämnts måste en långsiktig plan tas fram för appen. Då ett teknikval görs mellan att antingen utveckla native-appar eller använda sig av multiplattformslösningar blir det kostsamt att behöva ändra sig senare. Det arbete som har utförts på den tidigare plattformen måste till stora delar göras om. Vid valet av att utveckla native-appar behöver den långsiktiga planen inte vara lika bestämd då de möjligheter som finns med multiplattformsverktygen alltid är möjliga vid native-utveckling. Detta då native-teknikerna alltid ligger steget före. Ett exempel på hur tekniska begränsningar kan uppstå är till exempel om Fleet Management-sektionen väljer att utveckla en mobil webbapp där en karta med åkeriets fordon visas. Att utveckla en mobil webbapp som visar fordon på en karta är inget problem. Om sektionen däremot senare beslutar att lägga till ytterligare funktionalitet, exempelvis en funktion som låter trafikledaren skicka ett meddelande till föraren från appen, kan tekniska begränsningar finnas i att skicka meddelanden från en mobil webbapp. Därmed blir utvecklaren begränsad. Vid valet av att utveckla med hjälp av multiplattformslösningar blir det därför viktigt att ha en klar bild över appens slutgiltiga funktionalitet, vilket inledningsvis är svårt att åstadkomma.

Utveckling enbart en mindre del i processen

I hela processen från att besluta om att en app skall tas fram till att den lanseras är utvecklingsprocessen en mindre del. Tyngden kommer istället ligga på förstudien. För den första versionen av appen gäller det att ha klart för sig vilken inledande funktionalitet som appen skall innehålla och hur funktionaliteten görs användarvänlig. Den bästa användarupplevelsen erhålls vid användandet av native-appar vilket är ett argument som slår ett slag för native-appar. Med native-appar är det enklast att skapa plattformsspecifika användargränssnitt som användaren känner igen sig i. Det är även en av Nielsens (1994) tumregler. Dock kommer multiplattformslösningarna allt närmare en native-upplevelse och om appen är enkel går det att få en bra native-upplevelse även i en multiplatformsapp.

Gemensam kodbas

Den största fördelen med att använda sig av multiplattformslösningar är att det finns endast en kodbas för alla plattformar. Kostnaderna för utvecklingen av appen blir mindre då mindre specialistresurser krävs, särskilt om antalet intressanta plattformar skulle öka. Idag känns det dock fortfarande riskabelt att anamma dessa lösningar då det finns en osäkerhet på grund av att teknikerna inte är tillräckligt mogna. En ytterligare fördel med mobila webbappar skulle bli att sektionen för Fleet Management idag har en stor erfarenhet av att utveckla mot webb, vilket därmed skulle innebära att stora delar av kunskaperna inom området redan finns. Vid native-utveckling måste kunskap skaffas i att utveckla till varje enskild plattform. Oavsett valet av utvecklingsteknik måste dock kunskap skaffas inom gränssnittutveckling och användarvänlighet i mobila enheter.

Integrationsprocessen

Idag finns redan en server som förser webbportalen med information. Då de mobila enheterna kan hämta samma data och att Fleet Management-sektionen själv utvecklar tjänsterna på servern gynnar det integrationsprocessen. Då många intervjuobjekt har nämnt att just integrationsarbetet varit en process som tagit lång tid måste avdelningen vara redo att avsätta tid för anpassning av servrarna om så behövs vid utveckling av appar.

REST/SOAP

Angående protokollen REST och SOAP kan den utvecklade prototypen ansluta med var och ett av dessa protokoll. Det finns en tydlig skillnad i paketens storlek vid sändning och mottagning av data där REST har avsevärt mindre paket och där väntetiderna för användaren därmed även blir något mindre. Väntetidernas längd är generellt korta och påverkar därmed inte användarvänligheten även på en långsammare uppkoppling. Då en anslutning från mobila enheter med ett SOAP-protokoll fungerar och att paketens storlek vid dataöverföring är väldigt små skapas ingen motiverande faktor till att övergå från det existerande SOAP-protokollet till REST. En förändring i de bakomliggande systemen behövs därmed ej och ett eventuellt integrationsarbete kommer därmed ej försvåras på denna punkt.

7.5 Förvaltning och underhåll av app efter publicering

Förvaltning

Det finns idag intresse från flera håll på Scania om att utveckla appar till både interna och externa målgrupper. Det finns för tillfället en app publicerad av Scania på App Store och Android Market. För att underlätta både de administrativa och praktiska processerna mellan Scania och nätbutikerna rekommenderas att alla appar av Scania lanseras av ett gemensamt konto för att göra det enklare för både slutkunden och den egna organisationen. Slutkunden får en bra överblick över vilka appar Scania har lanserat under ett övergripande publiceringsnamn. Vid registrering av varje nytt konto måste en administrativ process genomgå vilket i sådant fall skulle kunna undgås. Att samma process måste genomgå gång efter gång är ett arbetsslöseri. Kontakten får istället ske mellan den förvaltande avdelningen och dem som önskar att utveckla en app och då det sker internt är kontakten lättare att sköta.

Appen efter publicering

Som tidigare nämnts kan inte hanteringen av appen släppas då en publicering har skett, detta då olika förändringar kommer behövas göras för att appen skall kunna fungera väl eller förbättras. De som kommer att använda appen kommer att ha åsikter om hur den fungerar. Denna återkoppling måste tas om hand och reflekteras över. Därmed borde en kanal finnas för hur användarna kan ge åsikter om hur

appen fungerar. Det är viktig information som måste samlas in. Dessa åsikter kommer utveckla appen på det vis som önskas av användarna.

Tidigare diskuterades att apparna som utvecklas till mobila enheter blir beroende av leverantörernas villkor mot plattformarna. De empiriska data som har erhållits visar att en kannibalisering sällan skett från webbplattformarna till apparna. Användaren har alltid möjligheten att använda sig av webbportalen istället för appen. Att förutspå om en kannibalisering skulle ske från webbplattformen till appen i fallet för Fleet Management är dock svårt. Användarens vanor kan snabbt förändras till att använda appen istället för webbportalen då appens tillgänglighet kan anses vara större. Därmed kan användarna bli mer beroende av appen. Om appen generellt inte fungerar är det inte heller någon bra publicitet då ett dåligt rykte sprids snabbt.

Förändringar sker då leverantörerna förändrar sina plattformar och apparna måste anpassas till dessa förändringar. För att dessa förändringar skall kunna ske måste resurser avsättas för att se efter när dessa förändringar sker och testa att apparna fortfarande fungerar som önskat. Det lanseras många enheter som använder sig av Android. Det är inte alltid apparna fungerar som tänkt på de nya enheterna vilket medför att appen måste testas på nya populära telefoner. Alla enheterna som har iOS lanseras av Apple. Då de även utvecklar iOS finns en bättre kompatibilitet av nya enheter.

Om appen skall utvecklas vidare eller buggfixning skall ske finns det skillnader i om appen utvecklas internt eller externt. Vid extern utveckling skapas ett beroende av en utomstående part. Om tjänsterna i appen blir mer kritiska kan en fördel ligga i att utveckla dem internt för att minska riskerna och beroendet. Detsamma gäller för efterlevnaden av appen då snabba förändringar kan behövas göras vilket medför att resurser måste snabbt finnas tillgängligt. Om dessa resurser finns internt blir problemet enklare att lösa. Det kräver dock att en levande kunskap finns om förändringarna som kommer att behöva göras innan problem med att appen inte fungerar uppstår. Organisationen behöver därmed reflektera över ifall det är möjligt att hålla sig uppdaterad inom ett nytt område.

8 Rekommendationer

I detta kapitel sammanfattas de rekommendationer som studien har kommit fram till.

Sektionen för Fleet Management rekommenderas att utveckla mobila appar för delar av de tjänster som idag finns på webbportalen. Förutom att organisationen blir en mer konkurrenskraftig aktör visar anammandet av nya tekniker upp att de hänger med i den tekniska utvecklingen. Denna faktor, tillsammans med observationen att appar idag är hett, kommer skapa en större synlighet av Fleet Management-tjänsterna både internt inom Scania och externt ut mot marknaden. Det hela skapar en mer modern bild av Scania. Appen kommer inte ge en direkt ökning av de monetära intäkterna men skulle kunna, i ett senare skede, vara en motiverande faktor till köpet av en högre tjänstenivå inom FMS. De monetära intäkterna är dock inte det primära i fallet utan den stora vinningen ses i ett mervärde för, och en förenkling av, åkeriets arbetssysslor vilket leder till en ökad lojalitet hos kunden.

Rekommenderat är att utveckla native-appar till iOS och Android. Viktigt är även att ha en långsiktig plan för appen som skall utvecklas. En vetskap om hur appen skall kunna användas med tanke på tekniker som till exempel kamera och GPS bör finnas redan i ett relativt tidigt skede. Dock borde omfattningen av funktionerna i den första versionen av appen vara grundläggande för att senare utvecklas i ständigt samråd med slutkunden. En långsiktig plan måste tas fram på grund av att valet av utvecklingsteknik kan begränsa de möjliga teknikerna i enheterna, detta gäller specifikt multiplattformslösningar. Valet av operativsystemen iOS och Android beror på att dessa plattformar är de som marknaden idag kräver för att nå ut till en bred publik. Windows Phone är en aktör som många tror på i framtiden och därmed måste ögonen hållas öppna för hur den utvecklingen går. Att inte utveckla med multiplattformslösningar motiveras av att de inte anses vara tillräckligt mogna än och därmed tas för stora risker om utveckling sker med dessa. Vid native-utveckling låser utvecklaren sig inte heller vid en strikt plan då de tekniska möjligheterna alltid är störst vid native-utveckling. En bättre användarupplevelse är också en motiverande faktor till native-utveckling. Dock krävs specialistkompetens för varje native-utvecklingsförfarande.

Idag finns det inga professionella mobilutvecklare på sektionen för REI. Det gör att kunskap om sådan utveckling måste tas in antingen genom utbildning eller att appens utveckling läggs ut på en extern leverantör. Det anses inte vara något problem att utbilda anställda utvecklare inom mobilutveckling då flera redan har visat ett intresse för och har kunskaper inom området. Om en extern leverantör tas in för utvecklingen rekommenderas det att företaget efter några iterationer eller versionssläpp fasar in utvecklingen internt för mer kontroll, lägre kostnader och minskade risker. Detta är särskilt viktigt om åkerierna uppvisar ett beroende av appen, vilket indikerar att appen i fortsättningen bör klassas som en kärntjänst. Det är också viktigt att i appens första skede skapa en arkitektur med modeller och struktur

på ett sådant sätt att appen skall kunna vidareutvecklas utan några större omvälvande förändringar i framtiden. Det gäller också att ha i åtanke vem som äger rättigheterna till koden.

Att ha tät kontakt med slutkunder har visat sig vara mycket viktigt vid apputveckling och en tydlig behovsbild är det mest centrala för att appen skall kunna bli framgångsrik. Vid utvecklandet av appen för produktion är det rekommenderat att ta med den återkoppling som fåtts av åkerier då den utvecklade prototypen presenterats. Det är även rekommenderat att erhålla återkoppling från kunder som inte är driftprovskunder för att få in ett brett och nyanserat spektrum av åsikter. Den primära funktionen som alla kontaktade åkerier vill se är en karta över fordonen och information om dessa fordon.

Inga konkreta hinder ses med att fortsätta använda Scantias SD-process och RUP-ramverket om de bestämmer sig för att bedriva apputvecklingen internt. Eftersom apputveckling bidrar med ett par särskiljande drag, bland annat ett stort fokus på interaktionsdesign och användbarhet och den väldigt viktiga återkopplingen mot slutanvändarna, kan dessa nya roller behöva representeras i ramverket. Dessutom rekommenderas det att fler medlemmar i utvecklingsprojektet, även utvecklare, får delta i kundkontakten då de olika rollerna inom apputveckling är så beroende av varandra. På grund av RUP-ramverkets omfattning kan det dock vara värt att titta på Scrum som utvecklingsmodell då den har använts framgångsrikt av intervjuobjekten och beaktar alla ovannämnda faktorer på ett mer förenklat vis. Att bedriva apputveckling efter Scrum-metoden kan påbörjas snabbt då metoden ej kräver någon särskild utvecklingspraxis. Som tidigare diskuterat lämpar sig Scrum för mindre team där flexibiliteten och den oförutsägbara omvärldsbilden är viktiga faktorer som bör beaktas.

Det är viktigt att fortsätta handha en app efter det att den publicerats. Nya versioner av de mobila operativsystemen och nya enheter gör att apparna måste underhållas för att fungera väl. Det går därmed inte att bara att överge appen när den publicerats. Att få in återkoppling och bearbeta den är en del som kan ske i samma process med underhållet mot nya versioner och enheter. Att sätta upp en kanal där användare av appen kan ge enkel återkoppling mot appen rekommenderas. Detta kan vara ytterligare en god anledning för att använda sig av Scrum-filosofin där utvecklingsprocessen ständigt ses som ett oförutsägbart moment.

Det protokoll som idag används för att hämta information från webbtjänsterna hos Fleet Management är SOAP. Då den utvecklade prototypen i studien har anslutit sig till dessa tjänster med SOAP-protokollet är utveckling av mobila appar inte en avgörande faktor för att implementera nya REST-protokoll för de existerande webbtjänsterna. Det fanns skillnader i paketstorlekarna mellan de olika protokollen men då det handlade om små skillnader blev användarupplevelsen ej påverkad. Utvecklingen blir enklare med REST-protokollet så en motiverande faktor för en övergång skulle vara ett förenklat arbete för utvecklarna, vilket leder till mindre slöseri.

Det rekommenderas även att alla framtida Scania-appar lanseras och förvaltas av en central organisation inom Scania. Detta skulle förenkla de administrativa processerna som annars måste genomgåas vid öppnandet av ett nytt konto varje gång en sektion väljer att utveckla en app. Istället kommer det att ske administrativa processer med avdelningen som önskar utveckla appar och den centrala organisationen men dessa processer anses ej bli lika problematiska med avseende på ledtider. Idag existerar redan all den publiceringsinformation som behövs för att Scania skall kunna publicera appar, men denna information bör vara lättåtkomlig. Intresse från flera håll inom Scania har visats för appar så med största sannolikhet kommer fler appar utvecklas av organisationen, vilket gör att denna utmaning kommer behöva bemötas.

9 Slutord

I detta kapitel presenteras en återkoppling till det syfte och de frågeställningar som inledningsvis presenterades. Dessutom presenteras en kort diskussion kring de förslag som författarna har på möjlig framtida forskning.

Att besluta om apputveckling skall ske eller inte kan vara ett större beslut än företaget först anar. Allt hänger på de utmaningar och förutsättningar som gäller för företaget i fråga, samt den potential som finns i appen om beslutet att utveckla den går igenom. Studien har kommit fram till att det i Scantias fall är rekommenderat att påbörja apputveckling. I inledningen av denna studie ställdes följande frågeställningar upp:

- *Vilka förutsättningar gäller för att Scania ska kunna ta fram en Fleet Management-app?*
- *Vilka organisatoriska och tekniska utmaningar finns vid utvecklingsförfarandet av mobila appar innan och efter lansering?*
- *Hur ska de utmaningar som finns beaktas för att en framgångsrik app ska kunna tas fram?*

Den första frågeställningen har behandlats genom att undersöka den miljö som Scania Fleet Management verkar i, bland annat med avseende på slutanvändarna, den behovsbild som existerar samt en analys av de konkurrenter som verkar inom Fleet Management-segmentet. I fallet med Scania utgår företaget från en etablerad och stabil teknisk plattform och det blir således ej ett stort steg att förlänga denna tjänst till att även innefatta den mobila sfären. Kostnaden för att ta fram en mobil applikation väger ej över de fördelar som erhålls. Då sektionen redan är starkt utvecklingsorienterad blir det lättare för organisationen att anpassa sig till nya tekniker, vilket leder till att riskerna minskar.

Den andra frågeställningen har behandlats genom att undersöka den rådande utvecklingsmodellen på sektionen, hur jämförbara företag har gått tillväga samt studerat de tekniska krav och begränsningar som råder i dagsläget. Inga hinder finns för att sköta utvecklingen in-house, men det är viktigt att beakta de nya roller, tekniska krav och utvecklingsfilosofier som apputveckling för med sig. Det gäller ej att underskatta det förvaltningsarbete som uppstår efter lanseringen av appen, särskilt om målgruppen ser appen som en viktig del i deras arbete. Studien har visat att vissa av de tekniska utmaningar som företaget ställs inför vid mobil utveckling har lösts genom att det idag finns ett gränssnitt som portalen använder sig av och som de mobila enheterna kan utnyttja. Ytterligare utmaningar är kunskapsinsamlingen av mobil utveckling, där olika alternativ har diskuterats i studien.

Den tredje frågeställningen sammanfaller till viss grad med de två tidigare frågeställningarna och besvaras genom att redogöra för vilka konkreta utmaningar, både organisatoriska och tekniska, jämförbara företag samt utvecklare har behövt bemöta för att nå framgångar, samt deras tillvägagångssätt. Slutanvändarens roll i apputvecklingsprocessen är den viktiga gemensamma

nämnamnaren både ur ett tekniskt och ett organisatoriskt perspektiv och tredjepartsutvecklare används som en snabb källa till specialistkompetens. Den existerande utvecklingsorganisationen behöver anpassas till mobil utveckling genom att använda sig av mer agila metoder. Genom att använda sig av kompetens från en extern leverantör, de egna erfarenheterna av branschen och mjukvaruutveckling samt en återkoppling från slutanvändarna skapas en god grund för att bemöta de utmaningar som finns inom apputveckling.

En reflektion och observation som författarna delar är det allmänna intresse som finns för appar inom hela organisationen. Om detta intresse leder till en allmän satsning på apputveckling blir det extra viktigt att tänka på hur utvecklingsförfarandet skall lösas och effektiviseras ur ett holistiskt perspektiv.

9.1 Framtida utveckling

Ur Scantias perspektiv har det visats ett stort intresse för en app som riktar sig mot chaufförerna av fordonen. Appen skulle kunna bli en ersättare av Interaktorn och dess funktioner såsom orderhantering, meddelanden och så vidare. Då chaufförerna visat intresse i att veta hur bra de kör med tanke på miljön och andra faktorer skulle appen kunna presentera information om just denna förare för att se hur de ligger till. Detta skulle i slutändan ge en bättre miljöaspekt då det visats sig att förarna vill förbättra sig. Detta kräver dock någon identifiering som inloggning av förarroll, vilket idag inte finns i Scantias Fleet Management-lösning.

På ett mer holistiskt plan kan det även vara av intresse att studera appen ur ett livscykelperspektiv med tanke på den raska teknologiska utveckling som sker inom detta fält. Apparnas och även de mobila enheternas snabba utvecklingstakt bör jämföras med klassiska mjukvaruutvecklingsprojekt som utvecklas under långa tidsperioder.

10 Litteraturförteckning

Böcker

Abrahamsson, Pekka et al. (2002). *Agile software development methods: Review and analysis*, Espoo 2002. VTT Publications 478.

Andersen, Bjørn & Pettersen, Per-Gaute (1997). *Benchmarking: en praktisk handbok*. Lund: Studentlitteratur

Bendell, Tony, Boulter, Louise & Kelly, John (1994). *Benchmarking for competitive advantage*. Göteborg: ISL (Institutet för säljträning och ledarutveckling)

Cockburn, Alistair (2007). *Agile software development: the cooperative game*. 2. ed. Upper Saddle River, NJ: Addison-Wesley

Collis, Jill & Hussey, Roger (2009). *Business research: a practical guide for undergraduate & postgraduate students*. 3. ed. Basingstoke: Palgrave Macmillan

Crinnion, John (1991). *Evolutionary systems development: a practical guide to the use of prototyping within a structured systems methodology*. London: Pitman

Eriksson, Lars Torsten & Wiedersheim-Paul, Finn (2006). *Att utreda, forska och rapportera*. 8., förnyade uppl. Malmö: Liber

Gibbs, R. Dennis (2006). *Project Management with the IBM Rational Unified Process: Lessons from the Trenches*. IBM Press

Karlöf, Bengt (2009). *Benchmarking: med lärande för att utveckla företag, organisationer och människor*. 1. uppl. Malmö: Liber

Kirk, Jerome & Miller, Marc L. (1986). *Reliability and validity in qualitative research*. Beverly Hills: Sage Publications

Kniberg, Henrik & Skarin, Mattias (2010). *Kanban and Scrum: making the most of both*. C4Media

Kruchten, Philippe (2000). *The rational unified process: an introduction*. 2. ed. Reading, Mass.: Addison-Wesley

Kvale, Steinar & Brinkmann, Svend (2009). *Den kvalitativa forskningsintervjun*. 2. uppl. Lund: Studentlitteratur

Poppendieck, Mary & Poppendieck, Tom (2006). *Implementing lean software development: from concept to cash*. Upper Saddle River, N.J.: Addison-Wesley

Schwaber, Ken & Beedle, Mike (2002). *Agile software development with Scrum*. Pearson international ed. Upper Saddle River, NJ: Pearson Education International

Sörqvist, Lars (2000). *Kundtillfredsställelse och kundmätningar*. Lund: Studentlitteratur

Vetenskapliga artiklar

Abbas, Rasha et al. (1998). Outsourcing Software Applications Development: Issues, Implications and Impact, *6th European Conference on Information Systems (ECIS'98)*, Aix-En-Provence

Dibbern, Jens et al. (2004). Information Systems Outsourcing: A Survey and Analysis of the Literature, *ACM SIGMIS Database*, Vol. 35, No. 4, s. 6-102

Golafshani, Nahid (2003). Understanding Reliability and Validity in Qualitative Research. *Test*, Vol. 8, No. 4, s. 597-606

Hall, Sharon P. & Anderson, Eric (2009). Operating Systems for Mobile Computing, *Journal of Computer Science in Colleges*, Vol. 24, No. 2, s. 64-71

Hirsch, Markus (2002). Making RUP Agile, *OOPSLA 2002 Practicioners Reports*, ACM

Johnstone, Stewart Dainty, Andrew & Wilkinson, Adrian (2009). Integrating products and services through life: an aerospace experience, *International Journal of Operations & Production Management*, Vol. 29, No. 5, s. 520 - 538

Larman, Craig & Basili, Victor R. (2003). Iterative and Incremental Development: A Brief History, *IEEE Computer Society*, Vol. 36, No. 6, s. 47-56

Nielsen, Jakob (1994). Heuristic evaluation. In Nielsen, J., and Mack, R.L. (ed.), *Usability Inspection Methods*, John Wiley & Sons, New York, NY

Quinn, James Brian (1999). Strategic Outsourcing: Leveraging Knowledge Capabilities, *Sloan Management Review*, Vol. 40, No. 4, s. 9-21

Quinn, James Brian (2000). Outsourcing Innovation: The New Engine Of Growth, *Sloan Management Review*, Vol. 41, No. 4, s. 13-28

Raddats, Chris (2011). Aligning industrial services with strategies and sources of market differentiation, *Journal of Business & Industrial Marketing*, Vol. 26, No. 5, s. 332-343

Schwaber, Ken (1995). SCRUM development Process, *Proceedings of the Conference on Object-Oriented Programming Systems, Languages, and Applications (OOPSLA'95) Workshop on Business Object Design and Implementation*

West, Joel & Mace, Michael (2010). Browsing as the killer app: Explaining the rapid success of Apple's iPhone, *Telecommunications Policy*, Vol. 34, s. 270-286

Elektroniska källor

Android Developers – Introduction (2012). Android Developers 2012-01-20, Tillgänglig: <http://developer.android.com/guide/developing/index.html> hämtad [2012-01-24]

Android Developers – Installing the SDK (2012). Android Developers, Tillgänglig: <http://developer.android.com/sdk/installing.html> [2012-01-16]

Android Developers – Platform Versions (2012). Android Developers 2012-01-03, Tillgänglig: <http://developer.android.com/resources/dashboard/platform-versions.html> [2012-01-04]

Android Developers – Preparing for Release (2012). Android Developers 2012-01-20, Tillgänglig: <http://developer.android.com/guide/publishing/preparing.html> [2012-01-24]

Android Developers – System Requirements (2011). Android Developers. Tillgänglig: <http://developer.android.com/sdk/requirements.html> [2011-10-17]

Android Developers – What is Android? (2012). Android Developers 2012-01-20, Tillgänglig: <http://developer.android.com/guide/basics/what-is-android.html> [2012-01-24]

Apple Developer – Apple Developer Program Enrollment (2012). Apple Developer, Tillgänglig: <http://developer.apple.com/enroll/selectEnrollmentType.php> [2012-01-24]

Apple Developer – Cocoa Application Competencies for iOS: Storyboard (2010). Apple Developer 2011-10-03, Tillgänglig: http://developer.apple.com/library/ios/#documentation/General/Conceptual/Devpedia-CocoaApp/Storyboard.html#//apple_ref/doc/uid/TP40009071-CH99 [2012-01-05]

Apple Developer – Cocoa Core Competencies: Delegation (2011). Apple Developer 2011-09-28, Tillgänglig: http://developer.apple.com/library/ios/#documentation/General/Conceptual/DevPedia-CocoaCore/Delegation.html#//apple_ref/doc/uid/TP40008195-CH14-SW1 [2012-01-18]

Apple Developer – Develop on your Mac (2012). Apple Developer, Tillgänglig: <http://developer.apple.com/programs/ios/develop.html> [2012-01-24]

Apple Developer – iOS Developer Enterprise Program (2012). Apple Developer, Tillgänglig:

<http://developer.apple.com/programs/ios/enterprise/> [2012-01-24]

Apple Developer – iOS Overview (2010). Apple Developer 2010-07-08, Tillgänglig:

http://developer.apple.com/library/ios/#referencelibrary/GettingStarted/URL_iPhone_OS_Overview/_index.html#//apple_ref/doc/uid/TP40007592 [2011-10-11]

Apple Developer – iOS Starting Point (2011). Apple Developer 2011-10-12, Tillgänglig:

http://developer.apple.com/library/ios/#referencelibrary/GettingStarted/GS_iPhoneGeneral/_index.html#//apple_ref/doc/uid/TP40007853 [2011-10-17]

Apple Developer – Tools Workflow Guide for IOS: About the Tools Workflow for iOS (2012). Apple

Developer 2012-01-09, Tillgänglig:

http://developer.apple.com/library/ios/documentation/Xcode/Conceptual/ios_development_workflow/_index.html [2012-01-24]

Apps to Go (2012). Masternaut, Tillgänglig: [http://www.masternaut.co.uk/vehicle-tracking/apps-to-](http://www.masternaut.co.uk/vehicle-tracking/apps-to-go.aspx)

[go.aspx](http://www.masternaut.co.uk/vehicle-tracking/apps-to-go.aspx) [2011-09-12]

Home - FleetBoard (2011). FleetBoard, Tillgänglig: <http://www.fleetboard.com/info/int> [2011-09-15]

Honan, Mathew (2007). *Apple unveils iPhone*, Macworld 2007-01-09,

<http://www.macworld.com/article/54769/2007/01/iphone.html> [2011-10-17]

HTML5 Tutorial (2012). W3Schools, Tillgänglig: <http://www.w3schools.com/html5/default.asp>

[2012-01-16]

Jokela, Sami (1999). *Short Introduction to Personal Digital Assistants*. Tillgänglig:

http://www.tml.tkk.fi/Opinnot/Tik-111.550/1999/Esitelmät/PDA/pda_final.htm [2011-11-28]

Kerris, Natalie & Bowcock, Jennifer (2009). *Apple's App Store Downloads Top 1.5 Billion in First*

Year, Apple 2009-07-14, Tillgänglig: [http://www.apple.com/pr/library/2009/07/14Apples-App-Store-](http://www.apple.com/pr/library/2009/07/14Apples-App-Store-Downloads-Top-1-5-Billion-in-First-Year.html)
[Downloads-Top-1-5-Billion-in-First-Year.html](http://www.apple.com/pr/library/2009/07/14Apples-App-Store-Downloads-Top-1-5-Billion-in-First-Year.html) [2011-11-28]

Kniberg, Henrik (2001). *Manifest för Agil systemutveckling*, Tillgänglig:

<http://agilemanifesto.org/iso/sv/> [2012-01-23]

Krazit, Tom (2004). *PDA Sales Hit A Slump*, PCWorld 2004-04-27, Tillgänglig:

http://www.pcworld.com/article/115879/pda_sales_hit_a_slump.html [2011-11-28]

MiX Mobile (2011). MiX Mobile, Tillgänglig: <http://mixmobileapps.com/> [2011-09-15]

Paczowski, John (2010). *The Mobile OS World: Symbian, iOS Are Superpowers; Android a Developing Nation*, AllThingsD 2010-12-01, <http://allthingsd.com/20101201/the-mobile-os-world-symbian-ios-are-superpowers-android-a-developing-nation/> [2011-10-18]

Pettey, Christy & Stevens, Holly (2011a). *Gartner Says PC Shipments to Slow to 3.8 Percent Growth in 2011; Units to Increase 10.9 Percent in 2012*, Gartner 2011-09-08, Tillgänglig: <http://www.gartner.com/it/page.jsp?id=1786014> [2011-10-11]

Pettey, Christy & Stevens, Holly (2011b). *Gartner Says Android to Command Nearly Half of Worldwide Smartphone Operating System Market by Year-End 2012*, Gartner 2011-04-07, Tillgänglig: <http://www.gartner.com/it/page.jsp?id=1622614> [2011-09-08]

Scania – *Fleet Management* (2009). Tillgänglig: <http://scania.se/tjanster/fleet-management/> [2011-09-05]

Scania – *Historia* (2009). Tillgänglig: <http://scania.se/om-scania/historia/> [2011-09-05]

Scania – *Kärnvärden* (2009). Tillgänglig: <http://scania.se/om-scania/scaniakoncernen/core-values.aspx> [2011-09-05]

Scania – *Scania in brief* (2011). Tillgänglig: <http://scania.com/scania-group/scania-in-brief/> [2011-09-05]

Scania årsredovisning 2010 (2011). *Årsredovisning 2010*, Tillgänglig: http://se.scania.com/Images/14582-Scania_SVE%2072%20indexerad_240607.pdf [2012-01-23]

TomTom *WEBFLEET Mobile - Highlight* (2011). TomTom, Tillgänglig: http://business.tomtom.com/en_gb/products/webfleet-mobile/highlights/ [2011-09-15]

Intervjuer

Andersson, Fredrik (2011). VD, Kondensator, 2011-09-21, Kondensators kontor, 80 minuter.

Andersson, Kenneth (2011). Managing Partner, The Mobile Life, 2011-10-13, The Mobile Lifes kontor, Stockholm, 50 minuter.

Axelsson, Mats (2011). Funktionsanalytiker, REIX, Scania, 2011-09-27, Scania Tekniskt Centrum, Södertälje, 40 minuter.

Bogeling, Jeff Thomas (2011). Trafikledare, LTD Spedition AB, 2011-11-18, Länna, 40 minuter.

Enelund, Kristofer (2011). Kravhanterare, REIS, Scania, 2011-09-15, Scania Tekniskt Centrum, Södertälje, 50 minuter.

Eriksson, Markus (2011). Funktionsspecialist, REIX, Scania, 2011-09-22, Scania Tekniskt Centrum, Södertälje, 60 minuter.

Goetzinger, Fredrik (2012). Kravhanterare, REII, Scania, 2012-01-17, e-postintervju.

Gustavsson, Magnus (2011). Field Test Engineer, Scania, 2011-10-25, 106C-10, Scania Tekniskt Centrum, Södertälje, 30 minuter.

Hallberg, Robert (2011). Sommarjobbare, Student vid Linköpings Universitet, 2011-09-02, Scania Tekniskt Centrum, Södertälje, 30 minuter.

Jansson, Joakim (2011). Trafikledare, Scania Transportlaboratorium AB, 2011-10-11, Scania Transportlaboratoriums kontor, Södertälje, 60 minuter.

Jansson, Joakim (2012). Trafikledare, Scania Transportlaboratorium AB, 2012-01-13, Scania Transportlaboratoriums kontor, Södertälje, 20 minuter.

Johansson, Carina V. & Funayama, Tina (2011). Försäljning, marknad & Systemansvarig, Taxi Stockholm, 2011-10-27, Taxi Stockholms huvudkontor, Stockholm, 40 minuter.

Johansson, Tobias (2011). Projektledare för mobila applikationer, SJ, 2011-11-02, e-postintervju.

Körberg, Patrik (2011). Försäljningschef för mobilen, Svenska Spel, 2011-11-22, Svenska Spel, Sundbyberg, 60 minuter.

Lundsgård, Anders (2011a). Systemarkitekt, REII, Scania, 2011-09-06, Scania Tekniskt Centrum, Södertälje, 60 minuter.

Lundsgård, Anders (2011b). Systemarkitekt, REII, Scania, 2011-09-08 Scania Tekniskt Centrum, Södertälje, 40 minuter.

Lundsgård, Anders (2012). Systemarkitekt, REII, Scania, 2012-01-09, Scania Tekniskt Centrum, Södertälje, 30 minuter.

Lundström, Elin & Tornblad, Rickard (2011). VD & interaktionsdesigner, Decuria, 2011-10-07, Decurias kontor, Stockholm, 40 minuter.

Nilsson, Marcus (2011). Business Development Manager - App and Mobile Banking, SEB, 2011-12-01, SEB Sergels Torg, Stockholm, 60 minuter.

Ramberg, Erik, Sommerstad, Kristoffer & Jepson, Jonas (2011). VD, utvecklare & anställd, SWAD (Springworks), 2011-09-26, SWADs kontor. 105 minuter. Jepson med efter 60 minuter. Sommerstad med under minuter 50-60.

Resare, Ingemar (2011). Transportrådgivare, Sveriges Åkeriföretag - ABC-åkarna, 2011-12-02, Sveriges Åkeriföretags kontor, 40 minuter.

Slott, Gunnar (2011). Projektägare av SVT Play, 2011-11-24, SVTs huvudkontor, 60 minuter.

Stjärnström, Frida (2011). Trafikledare, 2011-10-28, Stjärnströms International AB, 30 minuter.

Sunding, Curt, VD, CeEss Transport AB, 2011-10-20, Årsta partihallar, 30 minuter.

Watz, Peter, Lundsgård, Anders & Illanes, Christian (2011). InfoMate, systemarkitekt & systemutvecklare, Scania, 2011-10-04, Scania Tekniskt Centrum, 45 minuter.

Zandelin, Erica M. (2011). Manager, Internet Communications, Corporate Relations, Scania, 2011-11-02, telefonintervju, 30 minuter.

Åhlund, Roger (2011). Hyrbilsansvarig, Scania-Bilar Sverige AB, 2011-10-20, Kungens kurva, Stockholm, 20 minuter.

Intern dokumentation Scania

Eriksson, Markus (2011). *The Business of the Transport Company: A description of the context for Fleet Management Services*. Scania

Fleet Management - Mission, Vision & Strategy (2010). Scania

Hantering av krav REI (2010). Scania

Product Development Process (2011). Scania

R&D Factory (2010). Scania

Software Development Process, v 3.5 (2011). Scania

Bilaga – Intervjumallar

Då den kvalitativa undersökningen, i form av intervjuer, bestod av tre grupper av respondenter formulerade vi tre mallar för hur de semistrukturerade intervjuerna skulle gå till. De tre respondentgrupperna bestod av apputvecklingsföretag, användare av den nuvarande tjänsten samt jämförbara företag. Frågorna omformulerades eller raderades beroende på respondenten och avvikelser från mallen uppstod då det var befogat. Intervjuer som gjordes internt på Scania spelades ej in, och frågorna som ställdes var specifika för respondenten.

A - Mall för apputvecklingsföretag

Om företaget

- Berätta lite om företaget.
- Vad är din roll på företaget idag?
- Hur många utvecklingsprojekt har ni genomfört totalt?
 - Vilka appar har ni utvecklat?
 - Vilka projekt arbetar ni med just nu?

Utvecklingsprocessen

- Varför brukar kunderna vilja ha en app när de kommer till er?
- Vad använder ni för processer/metoder vid utveckling?
 - Varför använder ni dessa metoder?
 - Hur såg utvecklingsprocessen från start till slut ut för ett exempel?
- Hur ser tidsramen ut för ett vanligt apputvecklingsprojekt?
- Hur brukar rollerna se ut hos er? (generalister/specialister/rotation?)
- Hur brukar samarbetet med kunden se ut?
 - Hur brukar integrationssamarbetet se ut?
- Vilka delar av projekten brukar vara mest tidskrävande?

- Vad har ni för strategi för färdigutvecklade appar?
 - Hur hanteras rättigheter? (Vem äger koden?)
 - Vilken part underhåller appen efter lanseringsfasen?
 - Hur mycket jobb är det att registrera sig på App Store och Android Market som företag?
 - Hjälper ni företag att sätta upp/anpassa utvecklingsorganisationer/processer vid behov? (HR? Utvecklingsmiljöer?)
- Hur sker överlämningen av produkten? (exempel?)

Om olika plattformar

- Vilka olika plattformar har ni utvecklat appar till?
- Vad anser ni att de olika plattformarna har för för- och nackdelar?
- Hur brukar nedladdningssiffrorna se ut för era appar?
- Hur hanterar ni kundens önskemål om en applikation till flera plattformar?
- Vilket resonemang finns hos er angående om utvecklingen skall ske till plattformen direkt (native) eller om något verktyg skall användas för att utveckla till flera plattformar samtidigt?
 - Har ni stött på några specifika problem när ni har gjort detta vägval?
 - Vilka problem har uppstått?
 - Har det gjort att ni har fått byta tillvägagångssätt?
 - Om ni använder eller har använt några verktyg för utveckling till flera plattformar, vilka?
- Vad är era erfarenheter av att jobba mot web services?
 - Hur ser era kunder på att ha öppna API:er?
 - Hur ser ni på REST alternativt SOAP i en mobil miljö?
- Hur ser ni på säkerhetsaspekterna för appar?

B - Mall för driftprovskunder

Rutiner

- Hur ser arbetet för ert åkeri ut?
- Vad är din roll?
- Hur många fordon och förare har ni i dagsläget?
- Vilka är de generella roller som finns på ert åkeri?
 - Vilka roller använder FMP?
- Vilka är de rutiner som finns bland de olika rollerna?
 - Hur ser en typisk dag ut för dig?
- Finns det några sysslor i dina vanliga arbetsrutiner som skulle kunna förenklas av FMP?

FMP

- Vad är din inställning till Fleet Management-portalen?
- Hur ser användandet av Fleet Management-portalen ut idag?
 - Om förarna använder portalen, hur ser deras användande ut?
 - Vilka funktioner på portalen används?
 - Vilka funktioner används mest?
- Vilken erfarenhet har du av att använda mobila applikationer?
- Har du någonsin känt dig låst vid datorn eftersom FMP idag bara finns på webben?
 - Har du någonsin använt FMP via mobilens webbläsare?
- Känner du att någonting saknas i portalen?
- Hur skulle en mobil applikation påverka din syn av Scantias tjänst?

Förarna

- Hur ställer ni er till att förarna skall använda mobila appar inom sitt arbete?
- Använder förarna sina privata telefoner, eller har de tillgång till jobbtelefoner?
 - Om jobbtelefoner, vad är det för typ av telefoner? (enkla/smartphones?)
 - Skulle det komma på fråga att införskaffa telefoner åt era anställda om appfunktionerna visar sig användbara?
- Får förarna ta del av sin egen statistik från portalen på något sätt?
- Känner du att det finns ett behov av att öppna upp portalen för förarna?

Användning av portalen

- Noterar du de nyheter som idag finns på portalen?
- Hur används flotthistorik-funktionen?
- När du idag klickar på en bil på kartan, vilken teknisk information skulle du vilja ha ut och vilken teknisk information tittar du på idag på datorn?
 - Vilken av den här informationen skulle du vilja se i en mobil?
- Används portalen för att se underhållsplanering (mätarställningen) och andra variabler för ett fordon?
- Hur ser användandet ut av Traffic Light-funktionen (där förarna betygsätts)?
- Hur ser användandet ut av geofencing-funktionen?
- Använder ni er av larmfunktionen?
- Hur ser användandet ut av de sammanställningsrapporter som genereras?
- Hur brukar kontakten se ut mellan dig och föraren?
- Vilken av FMP-tjänsterna skulle du helst vilja se i en mobilapp?
- Om en mobil app för FMP införs, tror du den skulle få genomslag inom transportbranschen?

C - Mall för jämförbara företag

Uppstartsfasen

- Vad såg ni för värde i att skapa en app från första början?
- Vilken förståelse för appar fanns innan uppdraget påbörjades?
- Hur förankrades appen internt innan processen påbörjades?
 - Hur ser man på apparna internt? (marknadsföringsapp/verksamhetsnära app?)
- Finns det någon mjukvaruutvecklingsavdelning internt hos er?
- Är apparna gjorda internt hos er eller av tredjepart?
 - Vad är anledningen till det valet?

Utvecklingsfasen

- Vilka resurser och vilken tidshorisont tillsattes för uppdraget?
- Era appar finns till både iPhone och Android. Hur gick diskussionerna kring att skapa appen till dessa plattformar? (varierar beroende på appen)

Dessa frågor gäller om apparna är utvecklade av en tredjepart.

- Om ni utvecklar mjukvara/lösningar själva, varför lades uppdraget ut?
- Vad använde ni för kriterier när ni sökte efter en apputvecklare?
- Hur har processen och samarbetet sett ut med utvecklingsföretaget?
 - Vilken projektmodell har använts?
 - Hur iterativ har processen varit?
- Hur såg samarbetet ut när utvecklingsföretaget behövde ta del av det data som ni tillhandahåller?

Dessa frågor gäller om apparna är utvecklade inom det intervjuade företaget:

- Hur har utvecklingsprocessen sett ut?
- När den första appen skulle skapas för er verksamhet, fanns den nödvändiga kunskapen eller behövdes den införskaffas?
 - Inom vilka specifika områden kände ni att ni behövde mer kunskap, om den inte redan fanns?
- Har ni stött på några problem under utvecklingsfasen?
 - Om ja, vilka?

- Kunde appen integreras med era IT-system direkt, eller behövde förändringar genomföras dessförinnan?
- Hur ser nedladdningssiffrorna ut för de olika mobila plattformarna?

Tekniska frågor

- Skiljer sig iPhone-appen mot Android-motsvarigheten på något sätt? (om dessa plattformar)
- Hur ser ni på native-appar resp. multiplattformssappar i stort?
- Vilka säkerhetsrutiner har ni behövt tänka på för era appar?
- Använder er app sig av web services (REST eller SOAP) för att ansluta sig mot ett API?
 - Hur gick diskussionerna kring detta?
 - Var säkerheten något som påverkade det beslutet?
- Hur hanteras support mot kund?
 - Om apparna ej fungerar, vem har ansvaret för att återställa tjänsten?
- Vem ansvarar för den fortsatta utvecklingen av de existerande apparna (t.ex. uppdateringar och buggfixar)?
 - Om en extern leverantör av appar, hur ser förvaltningsavtalet ut?

Resultat och uppföljning

- Vilka konsekvenser, positiva som negativa, har appen bidragit med? (t.ex. större medial uppmärksamhet, högre lojalitet, förändrad syn på tjänsten, fler användare, starkare marknadsföring, klagomål om appen, olönsam investering, osv.)
- Riktat sig appen till existerande eller nya målgrupper?
 - Har ni noterat en ökning i antalet användare av era tjänster i helhet?
 - Har någon kannibalisering förekommit?
- Ser ni några problem med att externalisera ett appuppdrag i framtiden?
 - Ser ni någon vinning i att själva ta över fortsatt utveckling? (*om tredjepart just nu*)
- Blev resultatet som ni hade tänkt er?
 - Om inte, vad skiljde sig?
- Finns det en långsiktig plan för apputveckling i stort?
 - Hur ser den ut?
 - Hur har responsen varit från de verkställande organen?
- Vilka allmänna lärdomar har ni dragit av denna process?