



LUNDS UNIVERSITET

Ekonomihögskolan

Institutionen för informatik

Hantering av teknisk skuld inom den finansiella sektorn

En jämförelsestudie om hantering av teknisk skuld hos banker med olika teknologiska förutsättningar

Kandidatuppsats 15 hp, kurs SYSK16 i Informatik

Författare: Love Anderberg
Axel Eriksson
Oscar Wallin

Handledare: Magnus Wärja

Rättande lärare: Markus Lahtinen

Hantering av teknisk skuld inom den finansiella sektorn: En jämförelsestudie om hantering av teknisk skuld hos banker med olika teknologiska förutsättningar

ENGELSK TITEL: Technical debt management within the financial sector: A comparative study on the management of technical debt at banks with different technological qualifications

FÖRFATTARE: Axel Eriksson, Love Anderberg och Oscar Wallin

UTGIVARE: Institutionen för informatik, Ekonomihögskolan, Lunds universitet

EXAMINATOR: Markus Lahtinen

FRAMLAGD: maj, 2021

DOKUMENTTYP: Kandidatuppsats

ANTAL SIDOR: 91

NYCKELORD: Teknisk skuld, Bank- och finanssektorn, Teknisk skuldhantering, Förlegade system, Strukturomvandling

SAMMANFATTNING (MAX. 200 ORD):

Digitaliseringen har medfört en strukturomvandling som genomsyrar hela vårt samhälle. Det har lett till en teknologisk revolution för samtliga branscher, där spelplanen för aktörerna förändrats. Inte minst inom bankmarknaden där digitaliseringen medfört nya spelregler med PSD2 och Open Banking. Detta har ökat konkurrensen då många nya aktörer anträt marknaden med nya innovativa produkter och tjänster. Samtliga aktörer lägger således större fokus på utveckling, där exempelvis de mer traditionella aktörerna vill flytta affären från fysiska kontor till digitala applikationer. Detta kan medföra en ökning av teknisk skuld, eftersom utvecklingstempot ökar benägenheten att ta genvägar för att snabbare färdigställa utvecklingsprocessen. Undersökningen syftar därför till att besvara hur teknisk skuldsättning skiljer sig hos aktörer på bankmarknaden med olika teknologiska förutsättningar, samt hur skuldhanteringen ser ut hos dessa. Frågan är av induktiv karaktär och kvalitativ data har därför samlats in genom fem intervjuer med representanter från olika banker. Resultatet visar att de teknologiska förutsättningarna tycks skiljas mellan aktörerna och att det i sin tur har en påverkan över mängden skuldsättning. Bland annat påverkas skulden av faktorer som mängd legacy. Vidare visade det sig att hantering av teknisk skuld inte görs helt enligt metoder som teorin förordar. En del av strategierna fann vi gemensamma, men i allmänhet sker hanteringen mer informellt där samtliga tillfrågade företag tycks ha egna arbetssätt att hantera teknisk skuld.

Innehåll

1 INTRODUKTION	- 7 -
1.1 INLEDNING	- 7 -
1.2 PROBLEMMOMRÅDE	- 7 -
1.3 FRÅGESTÄLLNING	- 9 -
1.4 SYFTE.....	- 9 -
1.5 AVGRÄNSNINGAR	- 9 -
2 LITTERATURGENOMGÅNG	- 10 -
2.1 STRUKTUROMVANDLING	- 10 -
2.2 BANKMARKNADEN.....	- 11 -
2.2.1 Dynamiken förändras	- 12 -
2.2.2 Digitalisering av banksektorn.....	- 12 -
2.2.3 PSD2 och Open banking	- 14 -
2.2.4 Digitala banker.....	- 15 -
2.3 TEKNISK SKULD	- 16 -
2.3.1 Typer av teknisk skuld	- 18 -
2.3.2 Hantering av teknisk skuld	- 20 -
Identifiering	- 21 -
Mätning.....	- 21 -
Prioritering.....	- 21 -
Cost-Benefit Analysis	- 21 -
Portfolio Approach	- 21 -
Analytical Hierarchy Process (AHP)	- 22 -
Calculation of TD-principals.....	- 22 -
2.4 FÖRLEGADE SYSTEM	- 22 -
2.5 LITTERATURTABELL.....	- 24 -
3 METOD	- 25 -
3.1 METODVAL	- 25 -
3.2 URVAL	- 25 -
3.2.1 Val av organisationer.....	- 25 -
3.2.2 Val av intervjuobjekt	- 25 -
3.3 INTERVJUER.....	- 26 -
3.3.1 Tillvägagångssätt.....	- 26 -

3.3.2 Intervjuguide	- 27 -
3.4 BEARBETNING AV DATA	- 28 -
3.5 ETIK	- 28 -
3.6 VALIDITET OCH RELIABILITET	- 30 -
3.6.1 Validitet.....	- 30 -
3.6.2 Reliabilitet.....	- 30 -
4 EMPIRI.....	- 31 -
4.1 FÖREKOMST AV TEKNISK SKULD	- 31 -
4.2 TEKNISK SKULD OCH SYSTEM	- 32 -
4.3 TEKNISK SKULD OCH ARKITEKTUR	- 34 -
4.4 HANTERING AV TEKNISK SKULD.....	- 38 -
5 ANALYS & DISKUSSION	- 40 -
5.1 STRUKTUROMVANDLING	- 40 -
5.2 TEKNISK SKULD & FÖRLEGADE SYSTEM	- 40 -
5.3 TEKNISK SKULD & ARKITEKTUR	- 41 -
5.4 HANTERING AV TEKNISK SKULD.....	- 42 -
6 SLUTSATS	- 43 -
7 APPENDIX.....	- 45 -
7.1 INTERVJU 1.....	- 45 -
7.2 INTERVJU 2.....	- 53 -
7.3 INTERVJU 3.....	- 60 -
7.4 INTERVJU 4.....	- 67 -
7.5 INTERVJU 5.....	- 80 -
8 REFERENSLISTA.....	- 89 -

Figurer

Figur 1: Illustration av teknisk skuld (Fowler, 2019).....	- 17 -
Figur 2: Illustration på exempel hur teknisk skuld uppkommer (Sharma, 2018)	- 18 -
Figur 3: Förlegad applikationsstruktur (Warren 1999)	- 23 -

Tabeller

Tabell 1: Typer av teknisk skuld.....	- 20 -
Tabell 2: Sammanställning av litteratur	- 24 -
Tabell 3: Översikt av intervjuobjekt	- 27 -
Tabell 4: Intervjuguide.....	- 28 -
Tabell 5: Förekomst av teknisk skuld	- 32 -
Tabell 6: Teknisk skuld och system.....	- 34 -
Tabell 7: Teknisk skuld och arkitektur.....	- 38 -
Tabell 8: Hantering av teknisk skuld	- 39 -

1 Introduktion

1.1 Inledning

Digitaliseringen har under de senaste decennierna medfört en strukturomvandling som har genomsyrat samtliga delar av vårt samhälle. Denna teknologiska revolution har lett till att företag och andra organisationer har kunnat utveckla sina produkter och tjänster med hjälp av digitala medel, vilket vidare har lett till förändrade beteenden och högre förväntningar hos kunder. Kundernas högre ställda krav har även lett till att företagen ständigt tvingats bli allt mer innovationsrika och utveckla sina utbud för att konkurrera. Ett av de områden som kommit att påverkas mest av digitaliseringen, och som därmed har genomgått en stor förändring, är finanssektorn. Det är en sektor som under lång tid fungerat på ungefär samma sätt, där de fysiska kontoren varit centrala samtidigt som stordriftsfördelar och distributionsnät har premierats. De svenska storbankerna, som utgörs av SEB, Handelsbanken, Swedbank och Nordea, som numera är finskt, har därför haft en mycket stark och konkurrenskraftig position på marknaden under en väldigt lång tid, utan större behov av att göra fundamentala förändringar. Till exempel ägde de tillsammans i stort sett hela bankmarknaden under millennieskiftet år 2000, vilket exemplifierar deras dominanta ställning (Konkurrensverket, 2019). Emellertid har situationen förändrats i och med digitaliseringen av branschen, där de digitala möjligheterna och arbetssätten har gett upphov för nya regelverk och direktiv som ytterligare skyndat på den digitala strukturomvandlingen.

Liknande förändringar har kommit inom andra delar av finanssektorn, som till exempel för försäkringsbranschen, där konkurrensen har utvecklat sig på liknande sätt (Kevin Jiang, 2020). Den digitala utvecklingen har med andra ord gjort att de större finansiella aktörernas marknadspositioner börjat hotas något (Konkurrensverket, 2019). Exempelvis har nya finansteknologiska bolag (fintechbolag) bildats med digitala och innovationsrika produkter och tjänster som konkurrerar mot storbankernas. Dessutom skiljer sig de teknologiska förutsättningarna ofta väldigt mycket åt. Storbankernas system är ofta äldre vilket kan göra de svårare att vidareutveckla, medan de nyare aktörernas system är utvecklade senare och därmed modernare, ofta mindre komplexa och enklare att hantera (Dahlberg, 2021). För storbankernas del har det inneburit förlorade marknadsandelar inom flertalet områden, där mindre konkurrenter lyckats erbjuda mer specifika och lättanvända tjänster. Storbankerna har förvisso en fortsatt stark position inom ett antal områden, inte minst på ut- och inlåningsmarknaden, men denna strukturomvandling kan vara tecken på att marknadsandelarna kommer omfördelas ytterligare framöver.

1.2 Problemområde

I en undersökning av Accenture och SAP ansåg över 70% av anställda och chefer på 70 banker runt om i världen, att det största problemet med gamla banksystem är bristande flexibilitet. Många anser att konkurrenskraften påverkas negativt på grund av höga servicekostnader och dålig integration (Wallström, 2005). Vidare visade undersökningen att de svarande störde sig på de gamla arkitekturer och arbetssätt som användes och som i sin tur leder till bland annat inkonsekventa uppgifter i databaserna i de system bankerna använder sig av (Wallström, 2005). Men storbankerna arbetar hårt för att vidareutvecklas, exempelvis siktar Handelsbanken på att

dra ner sina bankkontor från 380 till 150 stycken, ett mål som de vill ska vara uppfyllt mot slutet av 2021. Detta görs i samband med att banken investerar en miljard kronor på IT-satsningar, vilket i sin tur kommer leda till uppsägningar för närmare 1000 anställda (Ekblom, 2020).

De senaste årens problematik inom banksektorn har alltså bestått av svårigheten för aktörerna att utvecklas och bli mer teknikbaserade (Borg, 2015). Lagar och regelverk har även förändrats för att anpassas till det nya digitala klimatet, vilket i sin tur har påverkat konkurrensfördelningen på marknaden och påskyndat den digitala utvecklingen. Till exempel har betaltjänstdirektivet *PSD2* tillkommit och skapat nya förutsättningar för samtliga aktörer på bank- och finansmarknaden (Finansinspektionen, 2019). Direktivet ger bland annat tredjepartsleverantörer rätt att tillgå information om en kunds bankkonto ifall kundens medgivande givits, oavsett vilken bank kunden har (Finansinspektionen, 2019). Denna förändring av transparens, vilken kommit att kallas för *Open Banking*, är ett resultat av flera andra förändringar som tillkommit kring öppen data och personuppgiftshantering hos företag. Det innebär att banksektorns aktörer har fått ökade skyldigheter, men samtidigt helt nya förutsättningar att utföra sina tjänster. Hur detta kommer påverka branschens olika aktörer är inte helt säkert. Vad lagen däremot gör, är att den tvingar branschen åt en digital strukturuomvandling, där alla aktörer i större utsträckning behöver digitaliseras för att förbli konkurrenskraftiga, vilket i sig ökar konkurrensen på marknaden.

Denna snabba utveckling har även lett till en del problematik hos aktörerna. Det finns nämligen en relation mellan ökad digitalisering och allt högre krav från kunder. Kunder vill ha så bra tjänster som möjligt, vilket företagen försöker uppnå med hjälp av nya, digitala och innovativa produkter. Samtidigt ökar kundernas krav i takt med att tjänsterna blir bättre och bättre. Detta kan likställas med en slags digital kapplöpning som i sin tur leder till växande och allt mer komplexa system (Lehman et al. 1997). Det här kan innebära en teknisk skuldsättning för organisationernas system, vilket i sin tur försummar flexibiliteten och ökar komplexiteten av dessa. Detta fenomen är även känt som *Teknisk skuld* (Fowler, 2019). Teknisk skuld skapar en problematik kring att hantera komplexitet och på ett lyckat sätt utveckla och förändra sig. Denna problematik skiljer sig åt beroende på vilken typ av aktör de infaller hos, då olika krav värdesattes för bankaktörernas grundsystem under olika tidsepoker (Bennett, 1994). De traditionella aktörernas affär uppkom till exempel långt innan ordet digitaliseringen var ett faktum, medan en del andra aktörer uppkommit som ett resultat av digitaliseringens möjligheter. Det föreligger således en problematik från grunden, där de teknologiska förutsättningarna skiljer sig mellan traditionella och moderna aktörer. Men vilka utmaningar och förutsättningar föreligger egentligen bankaktörerna för hantering av teknisk skuld? I denna rapport kommer vi undersöka hur olika typer av banker med olika teknologiska förutsättningar förhåller sig till och hanterar teknisk skuld.

1.3 Frågeställning

Hur skiljer sig teknisk skuld och hantering av denna åt mellan bankaktörer med olika teknologiska förutsättningar?

1.4 Syfte

Syftet med den här rapporten är att undersöka hur olika aktörer inom banksektorn hanterar teknisk skuld och jämföra hur deras teknologiska förutsättningar påverkar mängden av denna.

1.5 Avgränsningar

För att snäva åt den här rapportens undersökningsområde har vi valt att avgränsa oss till den svenska bankmarknaden. Problematiken gällande digitalisering, teknisk skuld och förlegade system förekommer i fler branscher, både inom och utom finanssektorn, men vi har valt bankmarknaden som undersökningsobjekt för den här uppsatsen eftersom branschen just nu befinner sig i en föränderlig fas. Teori- och empiriavhandlingen genom denna rapport kommer således vara applicerbar till flera aktörer inom finanssektorn. Vidare representeras de traditionella svenska storbankerna i rapporten av de fyra storbankerna SEB, Nordea, Handelsbanken och Swedbank, trots att Nordea numera har huvudsäte i Finland.

2 Litteraturgenomgång

Under litteraturgenomgången kommer vi först förklara begreppet strukturomvandling för att ge läsaren en förståelse av dess innebörd. Vi kommer sedan behandla bankmarknadens historia då det är av relevans att förklara hur sektorn har präglats av olika strukturomvandlingar de senaste decennierna. Detta för att få en förståelse för hur branschen är i ständig förändring och hur det påverkar de digitala aspekterna för företagen. Strukturomvandlingar kan förändra både spelreglerna och spelplanen för aktörerna och tvinga dessa till att anpassa sig efter nya digitala standarder och regelverk. Det här kan således leda till att företag måste lägga mer fokus på utveckling för att hinna med i de föränderliga omständigheterna, vilket i sin tur sätter högre press att nå deadlines, i synnerhet för systemutvecklarna. Teknisk skuld är i stora drag något som uppstår när utvecklare tar genvägar i sin utveckling, vilket försätter företag i en skuld som kommer behöva betalas tillbaka förr eller senare, och då med "ränta" eftersom det blir svårare att åtgärda ju längre tid det går. Således är ämnet Teknisk skuld något som blir särskilt relevant under strukturomvandlingar och andra föränderliga omständigheter i allmänhet. Därmed kommer vi ha en genomgång av de strukturomvandlingar som skett inom bankmarknaden de senaste decennierna fram tills idag och undersöka hur aktörerna påverkats av dessa. Vidare kommer vi in på teknisk skuld där vi avser beskriva vad fenomenet är, hur det skapas samt teorier med best practice för hur det bör hanteras. Slutligen kommer vi gå igenom begreppet *Förlegade system*, allmänt känt som *Legacy systems*, och förklara vad det är samt varför dessa ofta inte är anpassade till förändring. Eftersom vi ska undersöka hur hanteringen av teknisk skuld skiljer sig mellan företag med olika teknologiska förutsättningar, i synnerhet mellan företag med förlegade och nyare system, är det viktigt att skapa sig en förståelse för hur och varför system blir förlegade samt hur detta kan påverka hantering av både ny teknisk skuld och den skuld som redan föreligger sen tidigare.

2.1 Strukturomvandling

Samhället utvecklas i en ständig takt och genomgår alltjämt förändringar som på olika sätt påverkar samhällsstrukturen. Dessa förändringar kan vara olika betydelsefulla och därmed påverka samhället i olika grad. En typ av samhällsförändring som har en fundamental påverkan på samhället eller samhällsekonomin i stort, kallas för en strukturomvandling (Nationalencyklopedin, 2021). Genom historien har vi genomgått flera, både mer och mindre omfattande strukturomvandlingar. Till exempel då vi under slutet av 1800- och början av 1900-talet utvecklade nya industriella tekniker, vilket gjorde att vi frångick jordbruket och intog den industriella revolutionen (Nationalencyklopedin, 2021). Liknande samhällsförändringar har även skett under andra tidsepoker och omständigheter, bland annat på senare år då samtliga länder, branscher och individer har påverkats av digitaliseringen och de samhällsförändringar som det inneburit. En strukturomvandling kan således både utspela sig på olika sätt och bero på olika faktorer.

Globalisering, konkurrens, efterfrågan, regelverksförändringar och teknologisk utveckling är ofta faktorer som kan ligga till grund för dessa typer av strukturella förändringar. Ett tydligt exempel är bland annat då Apple gav sig in på mobiltelefonmarknaden för att konkurrera mot den dåvarande marknadsledande giganten Nokia. Apple insåg att den teknologiska utvecklingen hade kommit längre än vad marknadens produkter visade prov på, vilket möjliggjorde för Apple att skapa bättre och mer konkurrenskraftiga produkter (Doz & Wilson, 2018). Således

kunde de vinna marknadsandelar och överta Nokias roll som marknadsledare, något som de flesta ansåg omöjligt (Doz & Wilson, 2018). Strukturomvandlingen här handlar å andra sidan inte om att Apple faktiskt vann kampen och blev marknadsledare, utan om att den teknologiska utvecklingen, medhjälp av globalisering och förändringar i efterfrågan, har möjliggjort för mer konkurrenskraftiga produkter som i sig förändrat samhället på ett disruptivt sätt (Doz & Wilson, 2018). Med andra ord kan man säga att strukturomvandlingar både kan ses som en förutsättning för, och som ett resultat av, samhällstillväxt (Riksdagen, 2008).

Som framgår kan strukturomvandlingar uppkomma på olika sätt och därav bidra till olika konsekvenser. Vare sig det är samhällsliga förändringar som då vi frångick jordbruket för industrin, eller teknologiska förändringar som när Apple utvecklade smarta telefoner, så har förändringarna varit beständiga och legat som grund för nya förutsättningar och utmaningar. På liknande sätt kan man se en digital strukturomvandling inom banksektorn, där regelverksförändringar har lett till att allt fler digitala banker dyker upp på marknaden, vilket sätter större tekniska krav på de redan befintliga aktörerna (Eriksson, 2019).

2.2 Bankmarknaden

I december 2018 fanns 124 banker i Sverige (Svenska Bankföreningen 1, 2020). De kan traditionellt sett delas in i fyra huvudkategorier - bankaktiebolag, utländska banker, sparbanks och medlemsbanker, där bankaktiebolag är den största kategorin med 75 banker. Det totala antalet bankaktiebolag har ökat mycket under de senaste 10 åren, vilket är en följd av att flera svenska kreditmarknadsbolag ombildats till banker. Bankaktiebolag kan i sin tur delas in i tre kategorier - De svenska storbankerna, sparbanks som ombildats till bankaktiebolag, till exempel Swedbank-Sparbankerna och slutligen mindre bankaktiebolag med skiftande verksamhetsinriktning och ägarstruktur, som till exempel neobanks (Svenska bankföreningen 1, 2020). De svenska storbankerna brukade vara fyra, men är sedan Nordea flyttades till Finland tre stycken. Till de svenska storbankerna räknas således numera Skandinaviska Enskilda Banken (SEB), Handelsbanken och Swedbank. Storbankerna är vad man kallar för universalbanks, som kan erbjuda många finansiella tjänster och har tillsammans en stark position på den totala bankmarknaden, men andelarna varierar i olika delmarknader (Svenska bankföreningen 1, 2020).

Traditionellt har storbankerna tillgodosett behov som varit en förutsättning för de svenska hushållens inlåning. Under en tid då handel uteslutande skedde med fysiska kontanter, var lokala bankkontor en sådan förutsättning. Lokala bankkontor och personlig rådgivning är saker som stärkt den personliga relationen mellan hushåll och bank och därmed stärkt förtroendet för storbankerna. Av storbankerna är Handelsbanken och Swedbank de med flest kontor, 383 respektive 168. Eftersom de flesta banktjänster numera går att genomföra i mobiltelefonen eller på en dator är inte heller behovet av kontor lika stort som på 1990-talet, då telefon- och internetbanks först började ta andelar av inlåningsmarknaden. På inlåningsmarknaden har storbankerna tappat från cirka 80 procent under början av 2000-talet till 70 procent 2015 (Konkurrensverket 2018), och ner till 62 procent år 2019 enligt Svenska bankföreningen (Svenska bankföreningen 2, 2020). Även storbankernas fondbolag har tappat andelar på fondmarknaden från cirka 80 procent under tidigt 2000-tal till 55 procent 2018 (Konkurrensverket 2018). I och med denna förändring har storbankerna valt att avveckla många av sina kontor. Emellertid finns fortfarande vissa skäl att behålla en del av kontoren då de

tillhandahåller tjänster som personlig privat- och företagsrådgivning. Storbankerna har också fortsatt mycket administrativt arbete och utgör viktiga samhällsfunktioner som hantering av kontanter, vilket stärker behovet av fysiska kontor (Svenska Bankföreningen 2, 2020).

2.2.1 Dynamiken förändras

Storbankerna har en stark ställning på bankmarknaden i Sverige. Skälen till storbankernas starka position är enligt konkurrensverket inträdesbarriärer som stordriftsfördelar, inlåsnings effekter, infrastruktursamarbeten och regleringar (Konkurrensverket, 2018). Storbankerna kan erbjuda kompletterande tjänster och har ett stort kontorsnät, vilket har gett de en större kundbas och minskade distributionskostnader. Konkurrensverket skriver vidare att marknaden historiskt sett karaktäriseras av stordriftsfördelar. I takt med införandet av internetjänster och self-service-alternativ har dessa stordrifts- och nätverksfördelar emellertid avtagit vilket ökat konkurrensen på marknaden och försämrat storbankernas position. En annan aspekt som försämrat konkurrenssituationen för storbankerna är ökad rörlighet för hushållens placeringar. Eftersom hushållen svarar för cirka två tredjedelar av innehavet i fondmarknaden är deras möjlighet att flytta sina placeringar mycket viktig för konkurrensen. Införandet av investeringssparkonton (ISK), som möjliggör fondförsäljning utan reavinstbeskattning, har bidragit till att rörligheten i konsumenternas placeringar har ökat de senaste 15-20 åren och således är inlåsnings effekten inte lika stark. Detta har lett till ett mer utspritt sparande, med andra ord ett sparande som gynnar de banker som erbjuder smidig self-service (Konkurrensverket, 2018). På beställning av svenska bankföreningen har analysföretaget Copenhagen Economics gjort en kartläggning som visar på samma utveckling för konkurrensen. Copenhagen Economics skriver bland annat att dynamiken på den svenska bankmarknaden är större än i övriga EU. Detta kan förklaras av att inlåsnings effekterna blivit mindre för svenska bankkunder. Vidare är kostnaderna för ett bankbyte jämförelsevis låga, den höga grad av digitalisering möjliggör smidiga bankbyten samt att information som behövs för kreditbedömningar är allmänt tillgänglig vilket gör det enkelt och mindre osäkert för banker att ta emot nya kunder (Copenhagen Economics, 2019).

2.2.2 Digitalisering av banksektorn

Under de senaste decennierna har vår värld genomgått en digital transformation som har förändrat hela vårt samhälle. Denna digitaliseringsvåg har sköljt över alla våra marknader och branscher, inte minst banksektorn där Sverige är en av de mest digitaliserade i hela Europa (Svenska Bankföreningen, 2019). Vad som drivit på denna snabba förändring är inte bara den teknologiska utvecklingen i sig, utan även de förändrade kundbehov och regelverksförändringar som tillkommit i samband med denna utveckling. Före tiden med Internet skulle bankerna ha fysisk och lokal närvaro där deras kunder fick vända sig gällande bankärenden och dylikt. Idag förväntar vi oss istället att all information ska finnas tillgänglig direkt via mobilapplikationer, där vi även förväntar oss kunna hantera diverse finansiella tjänster utan problem. Bankernas kundprocesser och tillvägagångssätt såg med andra ord väldigt annorlunda ut i jämförelse med hur de ser ut idag, vilket som sagt även gäller för bankkundernas behov och förväntningar (Svenska Bankföreningen, 2016).

Digitaliseringen inom bankväsendet har inneburit flera olika förändringar som påverkat olika aspekter av den finansiella marknaden. Under 1970-talet började de traditionella bankerna med att automatisera och effektivisera sina processer och arbetssätt (Svenska Bankföreningen, 2016). Senare, i takt med att IT-vågen sköljde över världen under 1990-talet, började likaså internetbanker och nischbanker dyka upp på marknaden, till exempel Avanza som började erbjuda digitala tjänster för sparande och placeringar redan år 1999 (Avanza, 2021). Under 2000-talet fortsatte bankernas arbete med att förbättra sina produkter och utveckla sina digitala tjänster. Bankerna utnyttjade således digitaliseringens möjligheter och gynnades av bland annat effektiviserade processer och sänkta kostnader. Ett exempel på hur de utvecklade sina tjänster var då ett antal större banker i Sverige gick ihop för att skapa det digitala signeringsverktyget BankID, som implementerades i Sverige år 2003 (BankID, 2021).

IT och digitalisering stod fortsatt på tapeten under 2000- och 2010-talet och framfarten av nya finansteknologiska tjänster (fintech) fortsatte öka i markant takt. Storbankernas dominans i form av att kunna erbjuda de bästa banktjänster började ifrågasättas i takt med att nya finansteknologiska tjänster uppkom, men frågan var vilken inverkan det fått på konkurrensen på marknaden och om balansen faktiskt förändrats. I rapporten *Disruptiv digitalisering inom marknaden för juridiska tjänster*, utförd på uppdrag av Konkurrensverket år 2017, skriver författaren Christian Sandström om hur digitaliseringen påverkat marknaden för juridiska och finansiella tjänster. Sandström (2017) menar att bankerna hittills lyckats med sina informationstekniska investeringar och att de flesta finansiella ärenden faktiskt sker digitalt via internetbank eller bankens mobilapplikation. Samtidigt som de lyckats sänka personal- och lokalkostnader, menar Sandström att deras icke-teknologiska aspekter hållit sig relativt intakta. Han menar att bankernas varumärken behållit sin styrka, den lokala närvaron har fortfarande fungerat som inträde för nyblivna kunder samt även att de befintliga kundrelationerna inte tagit någon större skada. Vidare menar Sandström att bankerna fortfarande har en del stordriftsfördelar i jämförelse med mindre aktörer, till exempel gällande upplåning, vilket även det har medfört att de fortfarande har en dominant ställning (Konkurrensverket, 2017). Dessutom menar han att omfattande regleringar inom branschen sedan finanskrisen har skapat en väldig inträdesbarriär till marknaden som försvårat för uppstickare och nya aktörer generellt. Däremot fortsätter han med att det självklart funnits en del uppstickare som lyckats hålla sig kvar och som övertagit marknadsandelar från storbankerna. Främst handlar det om nätbankerna Avanza och Nordnet som med hjälp av sina användbara och enkla gränssnitt för digital aktiehandel lyckats överta markanta marknadsandelar inom aktiemarknadssegmentet (Konkurrensverket, 2017).

Förutsättningarna förändrades å andra sidan under 2010-talets andra hälft då ett antal större reformer inom banksektorn trädde i kraft, både nationellt i Sverige men även internationellt i Europa. I takt med att teknologier utvecklas och digitala verktyg implementeras i verksamheter och i organisationer, behöver även lagar och regelverk hänga på i utvecklingen och anpassas för vår digitala värld. Nya regelverk i form av PSD2, har tillsammans med moderna finansiella tekniker och tjänster som till exempel *Open Banking*, lett till en ökad tillströmning av aktörer på marknaden, vilket i sin tur har rubbat den traditionella värdekedjan som tidigare dominerats av de större finansiella instituten (Svenska Bankföreningen, 2019). Den digitala transformationen inom bankväsendet har med andra ord lett till en större omstrukturering av hela sektorn i stort. Nya nischbanker har tillkommit och likaså neobanker, en helt digital typ av bank utan fysisk närvaro. Dessa nya typer av banker och finansteknologiska tjänster har vuxit fram med hjälp av

digitaliseringens möjligheter, där de valt att fokusera på att förbättra specifika och mer nischade banktjänster på ett sätt som är svårt för bankerna att göra själva (Svenska bankföreningen, 2019). Dessa nya förutsättningar som digitaliseringen lagt grunden för i form av moderna teknologier och nya regelverk möjliggjorde således för andra aktörer att konkurrera mot storbankerna om samtliga delprocesser värdekedjan, istället för enbart slutprodukterna (Svenska bankföreningen, 2019).

Framfarten av digitala banker, finansteknologiska tjänster, nischbanker och neobanker, kan även förklaras genom den ökade digitaliseringen i allmänhet. Idag anses det inte i närheten lika okonventionellt att behandla viktiga handlingar över nätet som det ansågs för bara ett par år sedan. Dessa handlingar, som till exempel bankhandlingar, sjukhusjournaler och deklARATIONER, hanterades förr till största del i fysiskt pappersformat då motståndet mot det digitala var betydligt större. Hur det här synsättet har kunnat förändras så pass drastiskt på bara ett antal år beror självklart på flera olika faktorer. Utöver förbättrad och förenklad användarvänlighet hos aktörernas plattformar, kan det till exempel handla om att allt fler digitala tjänster strömmat in på våra marknader under denna period, vilket gjort att vi privatpersoner, tillika kunder, vant oss vid att använda dessa digitala tjänster i en mycket större utsträckning (Eriksson, 2019). Vidare kan det även förklaras genom att det skett ett antal teknologiska synergier som bland annat möjliggjort för implementering av artificiell intelligens och maskininlärning i dessa system, vilket har kunnat förbättra prestandan och även kunnat ersätta mycket av den mänskliga interaktion som tidigare krävts (Eriksson, 2019). Myndigheternas roll i spelet ska inte heller förbises när det kommer till orsaker för att allt fler fintech-lösningar existerar på marknaden. Först och främst har det varit tydligt att en del myndigheter ansträngt sig för att digitalisera sig, till exempel Skatteverket, Sjukvårdsupplysningen och Försäkringskassan, vars tjänster alltmer sker digitalt (Eriksson, 2019). Detta kan vara en bidragande orsak till att allmänheten lärt sig acceptera digitala medel i större utsträckning, även när handlingarna kan vara av känsligt slag. Historiskt sett har även myndigheter varit återhållsamma gentemot nya aktörer på finansmarknaden på grund av diverse riskfaktorer. Däremot har de, i takt med att fler aktörer tillkommit marknaden, behövt förädla sinkompetens gällande hur dessa aktörer ska kunna göra entré utan att säkerheten försummas, vilket även det gett slag i att fler aktörer numera existerar på marknaden (Eriksson, 2019).

2.2.3 PSD2 och Open banking

PSD2, förkortat från *Payment Service Directive 2*, är EU:s andra betaltjänstdirektiv och ersätter det tidigare direktivet PSD (Finansinspektionen, 2019). Finansinspektionens förändrade föreskrifter trädde i kraft i maj 2018. Därefter infördes kompletterande regler om bland annat starkare kundautentisering och kommunikation genom gränssnitt (API) i september 2019. Syftet med lagändringen är att utveckla och förbättra elektroniska betalningar och skapa förutsättningar för effektivare och säkrare betalningsmetoder. Lagändringen innebär ökade skyldigheter, men samtidigt förbättrade möjligheter för företag som tillhandahåller bankkonton samt tredjepartsleverantörer som tillhandahåller betalningstjänster. Kontoförvaltande institut har under särskilda omständigheter skyldighet att ge ut information om bankkonton till tredjepartsleverantörer, samt ge dem tillgång till kontot om kundens medgivande uttryckligen givits (Finansinspektionen, 2019). Med andra ord har leverantören alltså rätt att tillgå informationen och presentera den till kunden i fråga om denne begärt det. Detta möjliggör för vad som kallas *Open Banking*, ett samlingsnamn och en affärsmodell där tredjepartsbolag genom öppna APIer kan utveckla system och applikationer som använder sig av kontoinformation hos

andra kontoförvaltare (Premchand, Choudhry, 2018). Det innebär med andra ord att man kan koppla ihop ett system med ett annat och därmed få integrerade och mer sömlösa funktioner och tjänster. Open Banking kan därmed ses som en teknik för att på ett säkert sätt kunna ge företag, och således dess kunder, tillgång till information (Open Banking, 2021). Denna teknik baserar sig på förutsättningen och grundtanken av att kunderna själva äger den personliga och finansiella data som bankerna distribuerar över, och att man som ägare till datan därför har rätt att begära ut vilka leverantörer som ska få tillgå den (Deloitte, 2020).

Dessa förändringar innebär nya spelregler för samtliga aktörer och en ökad digitalisering av branschen som helhet, vilket lagt grunden för denna digitala strukturomvandling. Tredjepartsleverantörer kommer genom PSD2 kunna nå helt nya kunder eftersom kunderna numera kan kontrollera och få ta del av sina konton på olika banker, men i ett och samma gränssnitt. Under senare år har många nya aktörer slagit sig in på marknaden med betalningstjänster på just detta sätt, bland annat Apple Pay, Google Pay och Alipay. Alltså företagsjättar som utvecklar egna betalningsmetoder istället för att utnyttja mellanhänder. Genom denna utveckling ser vi en uppdelning på marknaden där företag och andra tredjepartsleverantörer kan utveckla sina tjänster kopplade till specifika delar av den finansiella tjänste- och produktmarknaden. Den traditionella finansiella värdekedjan, där hela processen hanteras av samma aktör, har därmed brutits upp och kan numera hanteras av flertalet olika aktörer, vilket ökar konkurrensen. Tidigare har istället hela värdekedjan dominerats av banken (Svenska Bankföreningen, 2019).

2.2.4 Digitala banker

Som tidigare nämnts har det tillkommit flertalet nya aktörer inom finansmarknaden som ett resultat av PSD2 och en mer transparent och öppen marknad. Den numera digitala bankmarknaden präglas dessutom av nya utmaningar och förutsättningar för aktörerna, inte minst gällande systemutvecklingen. Eftersom PSD2 och Open Banking öppnat upp för en mer öppen informationsutväxling mellan aktörerna, behöver även bankerna, tillika andra leverantörer, uppdatera sina utvecklingsmetoder för att dessa ska anpassas till den nya arkitekturen. Istället för att bygga stora system med många olika lager och funktioner, premieras det enligt den nya arkitekturen att bygga mindre och mer lätthanterliga applikationer med färre och mer koncentrerade funktioner, även känt som mikrotjänster (Deloitte, 2020). Detta beror på att det då enkelt går att integrera dessa mikrotjänster med varandra med hjälp av gränssnittskommunikation genom de öppna APIerna och på så sätt skapa kompletta och mer flexibla applikationer i en mikrotjänstarkitektur (Deloitte, 2020). Utvecklingen av mikrotjänster är snabbare och mer flexibel, vilket bland annat gör att uppdateringar kan släppas med jämna mellanrum istället för att ha stora kvartalsuppdateringar (Modig, 2018). Att bygga mikrotjänster kan å andra sidan vara svårt, i synnerhet om man ska förändra en redan befintlig och förlegad arkitektur, vilket ofta är förekommande hos gamla och stora organisationer som banker (Modig, 2018). Till exempel kan komplexiteten öka i takt med att antalet mikrotjänster ökar. Detta beror på att varje mikrotjänst måste vara integrerade med varandra för att funktionaliteten ska kompletteras, vilket skapar ett slags beroende mellan applikationerna (Modig, 2018). Däremot kan varje applikation hanteras, utvecklas och uppdateras individuellt, vilket gör de enklare att underhålla och vidareutveckla.

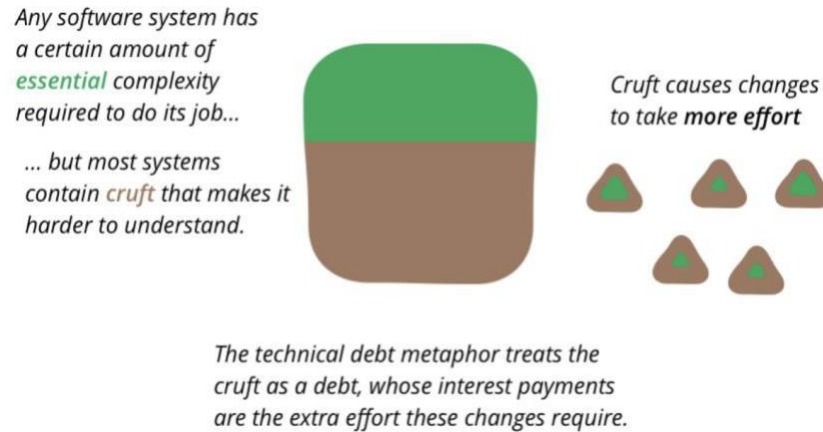
Dessutom kan de integreras sömlöst med andra applikationer och tjänster vilket gör att skalbarheten av dessa typer av system är betydligt högre. Mikrotjänster har således blivit ett

slags utvecklingsideal för de nya spelreglerna inom finansbranschen och tagit över de senaste åren som en modernare och mer effektiv utvecklingsmetod (Deloitte, 2020). Detta är något vi tydligt ser bland flera aktörer på marknaden, inte minst bland moderna finansteknologiska bolag och banker.

Finansteknologi, mer allmänt känt som fintech, är ett samlingsbegrepp för de finansteknologiska tjänster som tillkommit med hjälp av digitaliseringen inom finansmarknaden (Nicoletti, 2017). Med andra ord handlar det om att man med hjälp av digitala medel försöker förbättra och effektivisera traditionella finansiella tjänster, som till exempel betalningar, placeringar, sparande och kredithantering (Nicoletti, 2017). Det finns således flera olika typer av fintechbolag med olika affärsidéer och modeller, där en del vill underlätta individuella och specifika tjänster, medan andra vill digitalisera och förbättra hela koncept genom totallösningar. Bland dessa har vi neobankerna, ett relativt nytt uttryck som tagit rejäl fart de senaste åren i takt med att digitaliseringsvågen dragit över bank- och finanssektorn. Till skillnad från de traditionella storbankerna har neobankerna ingen fysisk eller lokal närvaro för sina kunder (Eriksson, 2019). Istället har de ett modernare och mer digitalt tillvägagångssätt, vilket visar sig då de enbart gör sig tillgängliga via digitala kanaler som till exempel mobil- och webbapplikationer (Eriksson, 2019). I Sverige har vi numera ett brett utbud av neobankerna med olika erbjudanden och anpassade finansiella tjänster som konkurrerar med storbankernas. Dessa tjänster handlar i stor mån om att förenkla individens privatekonomi genom att till exempel erbjuda smidiga tjänster för månads- och buffertsparande, betal- och kreditkort samt även lån och krediter i enkla, moderna och användarvänliga gränssnitt. Några exempel på neobankerna som är aktiva i Sverige är brittiska Revolut, svenska Rocker och P.F.C. (Kreditkoll, 2021).

2.3 Teknisk skuld

Mjukvarusystem är ofta benägna att skapa "cruft", eller brister i kvaliteten som försvårar modifiering och tillbyggnad av systemet. Detta fenomen, mer känt som Teknisk skuld är ett begrepp av Ward Cunningham som syftar till de konsekvenser som försvårar den framtida utvecklingen på grund av denna "cruft" (se Figur 1) (Fowler, 2019). Cunningham introducerade begreppet år 1992 och beskrev det med följande citat: "...shipping first time code is like going into debt..." (Tom & Vidgen, 2013). Med detta menade han att teknisk skuld går att likna vid en finansiell skuld (Fowler, 2019), där det extra arbete som krävs för att arbeta med de konsekvenser som kommer från dålig mjukvaruutveckling kan ses som en typ av ränta för skulden. Codabux & Williams (2013) förklarar teknisk skuld som något som uppstår till följd av snabba releaser vid mjukvaruutveckling och även om konsekvenserna av teknisk skuld är långsammare utveckling och försämrade effektivitet, finns det studier på att en liten dos teknisk skuld kan underlätta för utvecklare att snabba på utvecklingsprocessen kortsiktigt (Codabux & Williams, 2013).

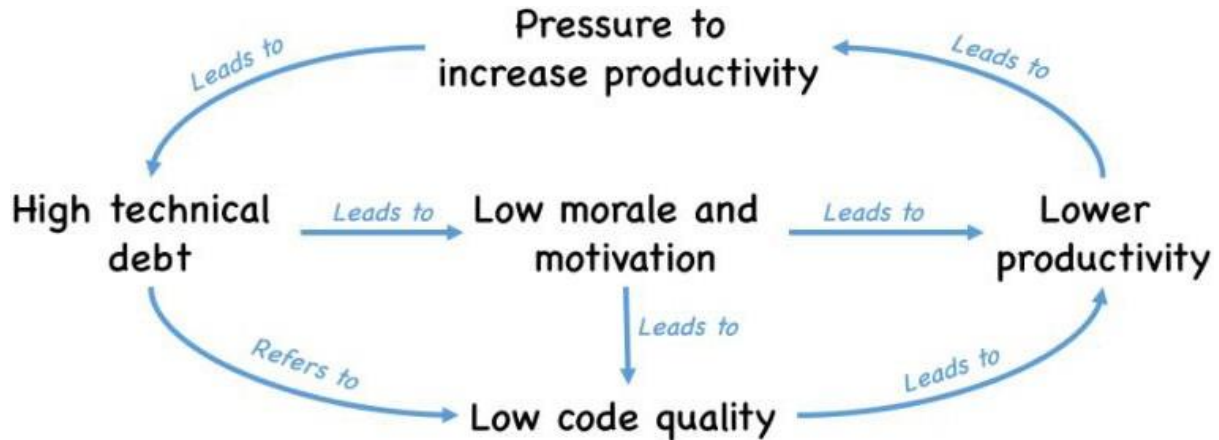


Figur 1: Illustration av teknisk skuld (Fowler, 2019)

Problematiken med teknisk skuld ligger inte i att eliminera den, utan istället handlar det om hur man bör handskas med den (Allman, 2012). Inom agila arbetssätt accepterar man att teknisk skuld är oundvikligt. Även om det är oundvikligt är det väldigt viktigt att denna tekniska skuld hanteras och åtgärdas snarast, då till exempel programmerare som känner till projektets tekniska skuld byter företag eller blir flyttade till andra projekt och på så vis lämnar efter sig en skuld som blir alltmer svår att åtgärda (Allman, 2012). Vidare menar Allman (2012) att det finns teknisk skuld som man bör undvika helt och hållet. Säkerhet är ett sådant exempel där genvägar kan leda till katastrof och där man inte har råd att stå i teknisk skuld. Man bör alltid föra journal över skulder som måste åtgärdas före release då det enligt Allman (2012) är väldigt lätt att glömma om man inte har det nedskrivet.

Det moderna arbetssättet *“release early and often”*, har gjort det allt enklare att angripa teknisk skuld. Om detta arbetssätt hanteras på rätt sätt och skulden behandlas tidigt, kan man göra släpp med högre frekvens, vilket vidare möjliggör tidigare feedback från användare eller i testning. Detta resulterar i en produkt som är mer anpassad till kunden. Om den tekniska skulden däremot inte åtgärdas i tid med detta arbetssätt leder det till att systemet kommer försämrats snabbare än vanligt (Allman, 2012). Allman (2012) menar vidare att all teknisk skuld inte är programmerares fel, utan att ledningen eller andra avdelningar också kan driva på den tekniska skulden. Inte minst då dessa inte har en lika initierad bild av skulden. Kunder tenderar att köpa funktioner och försummar den långsiktiga underhållningen av dessa funktioner. Därför brukar marknadsavdelningar skynda på processen och uppmuntra utvecklare att gå vidare till nästa projekt istället för att lägga tid på att förstärka, dokumentera och städa upp i det gamla systemet (Allman, 2012). Denna hets från andra avdelningar ger utvecklarna pressade tidsramar. Kod som är framtagen under hög tidspress tenderar att skapa mer teknisk skuld eftersom det inte är ovanligt att utvecklare då använder sig av hastiga metoder som fort färdigställer jobbet. Att kopiera kod som inte är direkt anpassad till systemet, eller att komma med snabba, illa designade lösningar, kan vara exempel där problemet till viss grad är åtgärdat, men skapar större och större komplexitet ju längre koden förblir. En skuld som inte hanteras inom skälig tid skapar en kedjereaktion som tenderar att landa i ytterligare teknisk skuld. Modellen nedan (Figur 2) illustrerar hur tidspressade implementationer skapar teknisk skuld och hur man genom att skjuta upp hanteringen av denna hamnar i en loop av skuld som kan vara väldigt svår att ta sig ur. Högt

teknisk skuld har sitt ursprung i slarvigt skriven kod, som leder till låg moral och lägre motivation, vilket leder till både lägre kvalitet och produktivitet, vilket i sin tur ökar tidspressen ytterligare (Sharma, 2018). Som tidigare nämnt av Allman (2012) leder detta till att systemet försämras snabbare, eftersom försämringen blir exponentiell. I händelse av att den upplupna skulden inte betalas eller åtgärdas i tid växer problemet och blir allt svårare att hantera. I extrema fall leder det till att mjukvaran hamnar i så kallad teknisk konkurs, vilket innebär att den blir i stort sett obrukbar. Det är då inte längre möjligt att införa ändringar på ett tillförlitligt sätt (Suryanarayana et al. 2015).



Figur 2: Illustration på exempel hur teknisk skuld uppkommer (Sharma, 2018)

2.3.1 Typer av teknisk skuld

Även om Cunningham beskriver teknisk skuld som ett problem avgränsat till koden och programmeringen (Fowler 2019), finns det många, såsom Alves et al. (2016) och Li et al. (2013), som menar att teknisk skuld kan återfinnas i fler delar av arbetet i den tekniska enheten. Alves et al. (2016) finner teknisk skuld i följande områden, sorterade i ordning på mest förekommande:

Designskuld	Skulden kan uppstå i relationer mellan entiteter/funktioner på konceptuell, logisk och fysisk nivå. Exempel är dålig praxis av kod, eller dålig design i relationer som i förlängningen skapar problematik på fysisk nivå.
Arkitekturskuld	Problem i arkitekturen ligger som grund för den tekniska skulden. Med arkitektur avses hur systemets olika moduler och komponenter är kopplade till varandra.
Dokumentationsskuld	Skulden uppstår i brist på felaktig eller saknad systemdokumentation.

Testskuld	Skuld som är en följd av icke kända eller kända problem i testningen som inte adresserats.
Kodskuld	Problem i källkoden som på ett negativt sätt påverkar möjligheten att underhålla koden. Lätt att identifiera genom att genomsöka koden efter dålig praxis.
Defektskuld	Kända defekter som bör adresseras men som skjuts upp och byggs därför upp till större skulder.
Kravskuld	En kompromiss som görs med olika implementationer som måste göras. En avvägning av implementationer där de avgörande implementationerna görs.
Infrastruktursskuld	En skuld i infrastrukturen som kan hindra eller försena vissa utvecklingsaktiviteter.
Personalskuld	En skuld som kan utgöra ett hinder eller försena utvecklingsaktiviteter. Ett exempel kan vara anställning av personer på fel plats eller inte tillräcklig utbildning och träning av de anställda.
Testautomatiseringsskuld	Avser arbete med att automatisera tester av tidigare utvecklad funktionalitet för att stödja kontinuerlig integration och snabbare utvecklingscykler.
Processskuld	Ineffektiva processer. Vad processen avser att hantera bör ta mindre tid än vad det faktiskt gör.
Byggskuld	Är en typ av kodskuld och avser problem som gör byggandet svårare och onödigt tidskrävande. Byggskuld kan innebära kod som inte bidrar med något värde för kunden. Om byggprocessen måste exekvera dåligt definierade beroenden blir processen onödigt långsam och osäker.
Serviceskuld	En skuld som liknar en infrastruktursskuld. Kan vara att man använder sig av webbtjänster som leder till bristande överensstämmelser mellan tjänstefunktionen och de krav som finns för applikationen.

Versionsskuld	Avser skuld som uppstår vid problematisk versionshantering av källkoden.
----------------------	--

Tabell 1: Typer av teknisk skuld

2.3.2 Hantering av teknisk skuld

Teknisk skuld kan alltså uppstå i olika delar av ett projekt såsom beskrivet ovan. Det kan endast hanteras på rätt sätt om samtliga delar av verksamheten har en förståelse för vilken inverkan den tekniska skulden kan ha i det långa loppet. Avvägningen mellan teknisk skuld och färdigställning av en produkt kan därför bara göras på rätt sätt om evaluering av problemet görs med god förståelse och hänsyn till skulderna nämnda i tabellen ovan. Med andra ord är det lika viktigt att även icke-tekniska avdelningar av en organisation har en förståelse för vilken inverkan teknisk skuld kan ha. Annars riskerar organisationen att lansera en produkt med för mycket skuld (Allman, 2012). Under föränderliga omständigheter likt de där en organisation genomgår omfattande systemrelaterade förändringar, blir det således extra viktigt att ha en god skuldhantering för att minimera ränta på skulden. När vi undersökt olika sätt att hantera teknisk skuld på har vi landat i två studier av Li et al. (2015) och Alves et al. (2016). Två studier som är bland de mest välciterade inom området. Det man kan se är att de till stora delar korrelerar och är överens i det sätt de förespråkar hantering av skulden. Gemensamt för studierna är att de utgår från en lista med åtgärder som bör tas för att minimera eller göra sig kvitt skulden. Vi har därför nedan presenterat hanteringen i de två studierna separat och sedan gjort en jämförelse av studierna, där vi finner att gemensamt för båda gällande hanteringen av teknisk skuld är att det läggs absolut störst fokus vid någon form av mätning, prioritering och återbetalning av skulderna. Båda studierna tar upp att kostnadsanalys är ett bra initialt tillvägagångssätt som kan ligga till grund för ordningen av vilken teknisk skuld som bör prioriteras. Båda studierna menar att dessa analyser bör göras för att bestämma vilka skulder som först bör betalas och när dessa ska betalas, eftersom de olika skulderna löper med olika så kallade räntesatser. Med detta menas egentligen att kostnaderna för att låta olika skulder vara obetalda är olika stora. De båda studierna listar metoder eller strategier med olika sätt att hantera teknisk skuld.

Li et al. (2015) har listat åtta olika metoder som de funnit är vanliga sätt att hantera teknisk skuld. I studien har de låtit förekomsten av metoden styra hur högt upp i listan den hamnar, dvs hur bra den kan anses vara. Metoderna som tas upp i listan är identifiering, mätning, prioritering, övervakning, återbetalning, kommunikation, förhindrande och representation/kommunikation. Alla metoder är enligt Li et al. (2015) olika sätt att angripa teknisk skuld, men för hantering av skulden bör dessa kombineras, så som till exempel identifiering, mätning, prioritering och återbetalning. Alves et al. (2016) beskriver på liknande sätt olika metoder för hantering. I studien nämns framförallt sex olika metoder, vilka är *Cost-Benefit Analysis*, *Portfolio Approach*, *Alternatives*, *Analytical Hierarchy Process*, *Calculation of TD-principal* och *Marking of Dependencies and Code Issues*. Båda studierna lägger som sagt störst fokus på mätning, prioritering och återbetalning. Det är också de tre metoderna som kommer vara av störst intresse för vår studie, då vi kollar närmare på om bankaktörer mäter, prioriterar och återbetalar teknisk skuld systematiskt. Vi har därför valt ut och djupare diskuterat metoderna identifiering, mätning och prioritering i studien Li et al. (2015) som beskrivs ytterligare nedan, samt Cost-Benefit

Analysis, Portfolio Approach, Analytical Hierarchy Process (AHP) och Calculation of Technical debt principals från Alves et al. (2016) som också beskrivs nedan. Mest korrelation finner vi mellan Portfolio Approach och identifiering, mätning och prioritering. Detta eftersom The Portfolio Approach är en strategi där skulderna prioriteras i en tabell efter bland annat uppskattad kostnad. Nedan följer en presentation av det material vi valt att utgå ifrån i studierna Li et al. (2015) och Alves et al. (2016).

Identifiering

Första steget är att organisationen bör identifiera skulden för att besvara frågan *“Vilken typ av skuld är det frågan om?”* Detta menar Li et al. (2015) kan göras genom till exempel analysering av källkoden, för att till exempel hitta överträdelser i programmeringsstandarder eller lösningar som är avvikande från praxis, och på något sätt påverkar produkten negativt.

Mätning

Med mätning av teknisk skuld menar Li et al. (2015) att man bör mäta kostnader för hantering av teknisk skuld. Detta bör göras genom estimerade värden för att åtgärda teknisk skuld, eller mäta skillnaden mellan nuvarande lösning och optimal lösning. Detta bör inkluderas vid budgetering och beräkning av projektet Li et al. (2015).

Prioritering

Genom att tidigare mätningar prioriterar man vilka tekniska skulder som bör åtgärdas först. Till exempel, om en åtgärd av en teknisk skuld leder till högre fördel än kostnad, ska detta åtgärdas så snart som möjligt. Vidare bör de tekniska skulder som är mest kostsamma åtgärdas först (Li et al. 2015).

Cost-Benefit Analysis

En utvärdering vars syfte är att avgöra om den förväntade effekten av att åtgärda skulden är tillräckligt hög för att motivera betalningen av skulden. Denna räntesats består av två delar: Den första delen består av sannolikheten att skulden, om den inte betalas, kommer att leda till extra kostnader för projektet. Den andra delen är uppskattad mängd arbete som krävs ifall skulden inte betalas (Alves et al. 2016).

Portfolio Approach

Den centrala delen för denna strategi är listan över poster för teknisk skuld. Denna lista innehåller krav som har identifierats för projektet. Registreringen sker i en tabell som innehåller skulden, den tidpunkt i vilken skulden uppkommit, vem som ansvarar för skulden, anledningen till varför den kan anses vara en skuld, en uppskattning av ränta, förväntad räntekostnad (EIA) och räntestandarsavvikelse (ISD), samt uppskattningar och jämförelser med andra poster för teknisk skuld. Under planeringen av varje steg i programvaran görs en analys om vad som ska betalas av och vad som ska skjutas upp (Alves et al. 2016).

Analytical Hierarchy Process (AHP)

AHP är en metod för att strukturera ett problem, jämföra alternativ med avseende på specifika kriterier och fastställa en övergripande rangordning för varje alternativ. Denna lista av varje alternativ appliceras på teknisk skuld och resultatet blir en ranking av skulder som identifierar vilka skulder som bör betalas först (Alves et al. 2016).

Calculation of TD-principals

Denna strategi fokuserar på de uppskattade kostnaderna av en given teknisk skuld. Syftet är att man genom att använda en kalkyl kan bestämma skuldsättningskostnaden och associera de identifierade problemen med olika kvalitetsstandarder som till exempel olika ISO certifieringar. Enligt författarna driver associeringen till olika kvalitetsstandarder cheferna att fatta mer rationella beslut kring vilka skulder som ska betalas först (Alves et al. 2016).

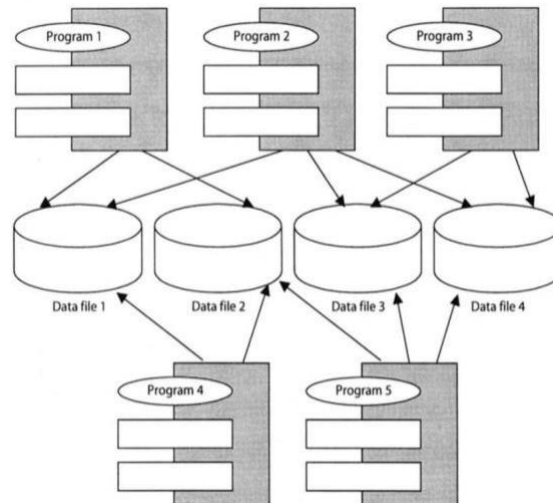
2.4 Förlegade system

Förlegade system är i all dagliga termer definierat som “ett stort system som vi inte vet hur vi ska handskas med men som är essentiell för organisationen” (Bennett, 1994). Ofta är dessa skrivna i ett tredjegerationsspråk såsom Coral, Fortran-66 eller Cobol (Bennet, 1994). Ett problem är att systemdokumentation ofta saknas, vilket gör systemen svårare att upprätthålla. För vissa system som hanterar vitala delar för en organisation kan felaktiga beslut sätta organisationens överlevnad på spel. Därför är många företag med förlegade system ofta ovilliga att förändra dessa (Bennet, 1994). Vidare försvårar föråldrad kodstruktur förändring och underhållning av dessa system (Bennett, 1994). Tidigare värdesattes effektivitet främst i systemutvecklingsprojekt.

Kostnaden av detta var att klarhet och struktur kom i andra hand och systemen utformades ofta med ad-hoc metoder, vilket är motsatsen till återanvändbar kod. På grund av detta är det idag också mycket svårare att förstå sig på förlegade system, speciellt i större organisationer där systemen kan innefatta hundratusentals rader kod. Att skapa sig en förståelse för sådana system är därmed mycket tidskrävande, vilket i sin tur leder till att det blir komplicerat att upprätthålla dem. Samtidigt är just systemförståelse en förutsättning för ett flexibelt och välanpassat system (Bennet, 1994). Tack vare systemförståelse kan man således skapa sig en bild av hur pass genomförbar en förändring är samt inse dess inverkan på resterande delar av systemet (Warren, 1999). Enligt Warren (1999) finns det tre anledningar till att majoriteten av förlegade system inte är designade för att hantera förändring:

1. *Kort förväntad livslängd.* När dessa system utvecklades hade man inte i åtanke att de fortfarande skulle vara i bruk decennium senare.
2. *Misslyckanden i processmodellering och mjukvaruutveckling gällande att hantera framtidens utveckling som en prioriterad aktivitet.* Det fanns krav på hur företag bör arbeta med utveckling, men dessa blev ofta ignorerade utifrån ett mjukvaruutvecklingsperspektiv.
3. *Begränsningar som existerade under systemutvecklingen.* Något som under senare år har blivit anmärkningsvärt billigare och mer kraftfullt är hårdvara. Vid utveckling av system för decennier sedan var minne och processorkraft några av de saker som påverkade utformandet av systemen. Detta har i sin tur lett till att dessa system har fått betala det dyra priset av att vara svåra att underhålla för att uppfylla dåtidens krav.

Figur 3 visar hur många förlegade systems applikationsstruktur ser ut. Figuren visar hur system består av flertalet olika program som använder sig av gemensam data. Implementationen av dessa system har ofta gjorts med användning av en så kallad platt-databas istället för ett "Database Management System" (DBMS). Det här leder i många fall till att datan blir duplicerad och därmed redundant då information lagras på olika sätt i olika filer. Detta försvårar förändringar i datastrukturen (Warren, 1999).



Figur 3: Förlegad applikationsstruktur (Warren 1999)

Vidare förklarade Warren (1999) även hur särskilda områden inom ett företag kan ha en negativ inverkan på system och dess funktionalitet, vilket kan leda till att de blir förlegade. Med andra ord aspekter som kan påverka systemets flexibilitet och fortsatta utveckling i allmänhet. Dessa aspekter är enligt Warren:

1. *Affärs mål.* Ett företags långsiktiga mål påverkar till stor del ett systems kravspecifikation. Desto radikalare affärs mål, desto mer resistent blir systemet mot förändring och anpassning till nya mål.
2. *Organisation.* En växande och "utvecklande" organisation är ansvariga för deras systems förmåga att hinna med. Precis som Bennett (1994) menar på, är detta väldigt svårt, då föråldrade system ofta saknar systemdokumentation. Vidare har individerna som haft hand om systemet börjat gå i pension vilket i sin tur leder till att organisationen får det allt svårare att underhålla sitt system.
3. *Teknik.* Från en teknisk synpunkt kan ett förlegat system brytas upp i applikationsmjukvara, systemmjukvara och hårdvara. En applikations kvalitet och skick, inkluderat dess systemdokumentation, är avgörande förutsättningar för mjukvarans möjlighet att anpassas och utvecklas. Vidare är en konstlad mjukvaruarkitektur en bidragande faktor till att förlegade system är svårföränderliga. I de fall där ett systems mjukvara inte längre är anpassat till dess mjukvara, är det oklokt att fortsätta investera i systemet utan att först byta ut mjukvaran.

2.5 Litteraturtabell

Kategori	Undersökningsområde	Litteratur
Strukturomvandling	<ul style="list-style-type: none"> • Strukturomvandling • Förändring • Teknologisk utveckling • Konkurrensituation • Inlåsningsseffekt • Self-service • Kundprocesser • Interna processer • Self-service • Innovation • Nya regelverk • PSD2 • Betaltjänstdirektiv • Open Banking • API • Betalningstjänster • Mikrotjänster 	<p>Nationalencyklopedin (2021), Yves Doz & Keeley Wilson (2017), Riksdagen (2008), Eriksson (2019).</p> <p>Svenska bankföreningen (2018, 2019, 2020), Konkurrensverket (2018).</p> <p>Svenska Bankföreningen (2019), BankID (2021), Avanza, (2021), Christian Sandström/Konkurrensverket (2017), Eriksson (2019).</p> <p>Finansinspektionen (2019), Svenska Bankföreningen (2019), Premchand & Choudhry (2018).</p>
Teknisk skuld & Förlegade system	<ul style="list-style-type: none"> • Cruft • Skuld • Tredjegerations-språk • Kodstruktur • Tredjegerations-språk • Kodstruktur • Dokumentation • Systemförståelse • Processmodellering • Begränsningar 	<p>Fowler (2019), Tom & Vidgen (2013), Codabux & Williams (2013), Guo & Searman 2011), Allman (2012), Sharma (2018), Suryanarayana et al. (2015), Bennett (1994), Warren, (1999).</p>
Teknisk skuld & Arkitektur	<ul style="list-style-type: none"> • Kodstruktur • Tredjegerations-språk • Mikrotjänstarkitektur 	<p>Bennett (1994), Warren, (1999), Deloitte (2020), Modig (2018).</p>
Hantering av teknisk skuld	<ul style="list-style-type: none"> • Tillvägagångssätt • Cost-Benefit • Approach • Mätning • Prioritering • Identifiering 	<p>Alves et al. (2016), Li et al. (2013).</p>

Tabell 2: Sammanställning av litteratur

3 Metod

I följande kapitel går vi igenom vald metod för uppsatsen. Vi börjar med att förklara vilken typ av metodval vi gjort för att sedan beskriva de företag och personer vi intervjuat. Därefter förklarar vi hur intervjuerna genomfördes för att slutligen komma in på hur den insamlade datan bearbetades samt dess validitet, reliabilitet och etik.

3.1 Metodval

I valet mellan kvalitativ ansats och kvantitativ ansats för insamling av empiriskt material valde vi att ta en kvalitativ ansats då vår forskningsfråga är av induktiv karaktär och kräver en djupare förståelse för hur intervjuobjekten arbetar med teknisk skuld (Jacobsen, 2002). Därför behöver vi djupare och mer ingående data vilket just kvalitativa studier ger, enligt Bryman & Bell (2015). Till följd av detta har antalet respondenter begränsats till fem. Ifall mer tid givits hade fler intervjuobjekt kunnat tillfrågas och bredden av respondenter ökats. Vi ville få en djupare insikt till organisationers relation till teknisk skuld och hanteringen av denna. Därför ansåg vi även att vi genom intervjufrågor kunde få mer värdefulla svar. Möjligheten att kunna gå utanför de ordinarie frågorna och ställa frågor som är mer anpassade åt den ifrågavarande respondenten är också bidragande till metodvalet.

3.2 Urval

I följande kapitel kommer vi beskriva valen av organisationer och senare även valen av personer att intervjua inom organisationen.

3.2.1 Val av organisationer

Då vi ville undersöka hur olika aktörer med olika digital mognad inom bankmarknaden hanterar teknisk skuld behövde vi ha en diversifiering hos organisationerna vi intervjuade för att få en nyanserad diskussion. Genom att både kontakta traditionella och nyare aktörer resulterade det i att vi fick intervjua med två aktörer med äldre system och tre aktörer med modernare system. Således kunde vi samla in empiriskt material från olika aktörer som dessutom har olika teknologiska förutsättningar för att sedan jämföra likheter och skillnader i deras hantering av teknisk skuld.

3.2.2 Val av intervjuobjekt

Val av intervjuobjekt tog rum i samband med val av organisation, då vi utgått från kriterier som innefattar både att intervjupersonen ska vara från en organisation som är verksam inom svensk banksektor, med det avser vi även fintechaktörer och neobanker. Utöver det är ett kriterium att intervjupersonen arbetar med och är väl initierad i organisationens tekniska omfång på en nivå som frågeställningen kräver. Slutsatsen av detta var att vi letade efter medarbetare som antingen direkt arbetade med den tekniska skulden, eller var väl initierad i den tekniska verksamheten och

i synnerhet införstådda i hanteringen av teknisk skuld. Därför ansåg vi att personer med IT-relaterade chefspositioner var av intresse för studien. Utöver dessa chefspositioner ansåg vi att det var av likvärdigt intresse att intervjua utvecklare för organisationerna, eftersom de på daglig basis arbetar med utveckling och har ofta god förståelse för systemet samt den tekniska skulden. Av den anledningen ansåg vi att den typen av intervjuobjekt också var av högt intresse. Kontakt med intervjuobjekten har primärt skett genom sociala affärsnätverk på internet, där intervjuobjekten blivit direkt tillfrågade av oss genom privatmeddelanden. Två av intervjuobjekt har erhållits genom personlig anknytning till vederbörande.

3.3 Intervjuer

Insamlingen av det material som ligger till grund för besvarandet av vår forskningsfråga skedde med metoden intervju. Som tidigare nämnt gjorde vi denna avvägning eftersom kvaliteten och djupet av intervjuerna är en förutsättning för att besvara vår forskningsfråga, då intervjuer enligt Jacobsen (2002) är den bästa metoden för besvarande av kvalitativa forskningsfrågor. Intervjufrågorna utformades med hjälp av en intervjuguide, bestående av fem olika huvudområden inom vilka vi utformat frågor kopplade till olika teorier i teorigenomgången. I intervjuguiden har vi även tagit med vissa följdfrågor för att få mer beskrivande svar från respondenten, samt utformat frågorna på ett sätt som ger plats till följdfrågor som uppstår under intervjun. Detta för att ge en dynamisk intervju och ett ytterligare djup som bättre besvarar frågeställningen.

3.3.1 Tillvägagångssätt

Vi ansåg att det var av hög vikt att vi fick intervjua respondenterna muntligen och i realtid. Detta eftersom det ger en mer dynamisk intervju, underlättar för följdfrågor och förtydligar kommunikationen mellan oss och respondenten, precis som Bryman & Bell (2015) menar på är en semistrukturerad intervju. Genomförandet av intervjuerna har gjorts genom programvaran Zoom som är en programvara för videokonferenser och digitala möten. Under normala omständigheter hade det varit möjligt för oss att genomföra intervjuer där både vi och respondenten närvarar fysiskt. Den fysiska närvaron hade eventuellt kunnat ge en mer raffinerad intervju med alla de effekter det har för samtalet. Digitala intervjuer blev emellertid det enda tänkbara valet för oss i och med den uppenbara problematik som råder kring fysisk närvaro och smittspridning av Covid-19. Dessutom har ett digitalt tillvägagångssätt möjliggjort intervjuer som annars hade varit svårare att genomföra på grund av rent geografiska skäl. Samtliga uppsatsskrivare och respondenter har därför befunnit sig på olika platser vid intervjun. Telefonintervju eller annan likvärdig metod var också möjlig men eftersom samtliga respondenter hade vana vid Zoom så genomfördes intervjuerna på det viset.

Organisation	Namn på anställd	Roll inom företaget	Typ av intervju	Längd	Inspelad
Avanza	Joakim Sahlström	CTO (Chief Technology Officer)	Zoom	29 minuter	Ja

Rocker	Oscar Alvarsson	CTO (Chief Technology Officer)	Zoom	30 minuter	Ja
Anonym (Svensk storbank)	Anonym	Software Developer	Zoom	28 minuter	Ja
Anonym (Svensk storbank)	Anonym	Senior kravanalytiker	Zoom	75 minuter	Ja
Carnegie Investment Bank	Lennart Lindholm	CISO (Chief of Information Security)	Zoom	32 minuter	Ja

Tabell 3: Översikt av intervjuobjekt

3.3.2 Intervjuguide

Vår intervju var uppdelad i fem olika avsnitt. Första avsnittet var introduktionsfrågor där intervjuobjektet fick en möjlighet att berätta om sig själv och sin professionella bakgrund för att sedan komma in på allmänna frågor om hur digitaliseringen påverkat banksektorn rent tekniskt och om deras syn på PSD2. För att sedan snäva åt intervjun mer mot vår forskningsfråga behandlade nästa avsnitt frågor om teknisk skuld hos organisationen, vilken typ som förekommer samt hur denna behandlades. Avsnittet efter teknisk skuld går mer in på företagets system och tekniska förutsättningar samt dess koppling till teknisk skuld. Sista avsnittet handlar om mikrotjänster, då det är en arkitektur som blivit allt vanligare hos branschens aktörer.

Område	Fråga	Koppling till teori
Introduktionsfrågor	<ol style="list-style-type: none"> 1. Vad heter du? 2. Vad är din roll och arbetsuppgift på företaget? 3. Vad har du för arbetsbakgrund? 	
Strukturömvandling	<ol style="list-style-type: none"> 4. Hur skulle du beskriva att digitaliseringen påverkat konkurrensen på bankmarknaden under de senaste åren? 5. Har PSD2 och Open banking förändrat era förutsättningar rent konkurrensmässigt? <ol style="list-style-type: none"> a. Om ja: <ol style="list-style-type: none"> i. Har det haft positiva effekter för er? ii. Har det haft negativa effekter för er? b. Har dessa förändringar gjort att ni ändrat fokus/strategier eller arbetar ni på liknande sätt som tidigare? <ol style="list-style-type: none"> i. Om ja: <ol style="list-style-type: none"> 1. Har dessa förändringar varit avgörande för att ni fortsatt ska vara konkurrenskraftiga? Strunta i denna? 	<ul style="list-style-type: none"> • Strukturömvandling (kap 2.1) • Bankmarknaden (kap 2.2)
Teknisk skuld	<ol style="list-style-type: none"> 6. Känner du till begreppet teknisk skuld? <ol style="list-style-type: none"> a. (Om nej, förklara kortfattat begreppet). <ul style="list-style-type: none"> - Kan liknas vid en finansiell skuld - Genvägar som tas pga snabb färdigställning pga tex press från omgivningen - Kostnaden som uppstår av dessa genvägar, ful kod, dålig arkitektur etc. 7. Är er verksamhet påverkad av teknisk skuld? 	<ul style="list-style-type: none"> • Teknisk skuld (kap 2.5) • Typer av teknisk skuld (kap 2.5.1) • Hantering av teknisk skuld (kap 2.5.2)

	<ol style="list-style-type: none"> a. Om ja: <ol style="list-style-type: none"> i. Kan du beskriva denna skuld? 8. Vad består skulden av? 9. Hur uppkommer skulden? 10. Hur arbetar ni med att hantera teknisk skuld? Använder ni er av några särskilda strategier för att hantera det? Har ni rutiner och egna best practices för det? <ol style="list-style-type: none"> a. Om ja: <ol style="list-style-type: none"> i. Kan ni berätta lite om dessa strategier? ii. Tex, arbetar ni med någon form utav prioritering av skulderna? Alltså värderar ni skulderna mot varandra? 11. Kan ni se en koppling mellan teknisk skuld och komplexiteten på era system? <ol style="list-style-type: none"> a. Om ja: <ol style="list-style-type: none"> i. På vilka sätt? 12. Kan ni se att tidspress från andra avdelningar leder till ökad teknisk skuld? T.ex att en annan avdelning pressar tidsramen då de inte har en förståelse för utvecklingsprocessen. <ol style="list-style-type: none"> a. Om ja: <ol style="list-style-type: none"> i. På vilka sätt? 13. Finns det något ni borde ändra på gällande hanteringen av teknisk skuld? 	
Teknisk skuld, förlegade system & arbetsmetoder	<ol style="list-style-type: none"> 14. Vad har ni för olika system och hur är de kopplade till varandra (Grovtt beskrivet)? 15. När är era äldsta system ifrån? (Som fortfarande är i bruk) 16. Vilket/vilka programmeringsspråk är ert/era system skrivna i? <ol style="list-style-type: none"> a. Finns det en koppling mellan hur gammalt systemet är och teknisk skuld? 17. Arbeta ni med agila arbetsmetoder? <ol style="list-style-type: none"> a. Om ja: <ol style="list-style-type: none"> i. Kan ni se en koppling mellan att det agila arbetssättet underlättar att hantera växande komplexitet av systemen? 18. Kan ni se att digitaliseringens utveckling har påverkat vad ni värdesätter i era system? Tex effektivitet, flexibilitet eller användbarhet. 	<ul style="list-style-type: none"> • Förlegade system (kap 2.7)
Teknisk skuld & arkitektur	<ol style="list-style-type: none"> 1. Använder ni er av mikrotjänster? 2. Om ja: <ol style="list-style-type: none"> a. Varför? b. Ser ni några nackdelar med användandet av mikrotjänster? c. Ser ni någon relation mellan hantering av teknisk skuld och mikrotjänstarkitektur? 3. Om nej: <ol style="list-style-type: none"> a. Varför? b. Strävar ni efter en mikrotjänstarkitektur (MSA)? 	<ul style="list-style-type: none"> • Digitala banker (kap 2.2.4)

Tabell 4: Intervjuguide

3.4 Bearbetning av data

Samtliga intervjuer tilläts att spelas in av intervjupersonerna och spelades in med både ljud och bild. Detta underlättade sedan för vårt arbete att transkribera dessa. Då intervjuer ger väldigt detaljrika svar (Bryman & Bell, 2015) är det viktigt att kunna återspegla dessa senare i transkriptionen. Vidare menar Bryman & Bell (2015) att transkriptionen är viktig då det bidrar till en transparens av det insamlade materialet vilket ökar dess reliabilitet då författarna således inte kan blanda in subjektivitet i svaren. Transkriptionen ger också läsaren möjlighet att utvärdera intervjuerna för att själva avgöra validiteten och reliabiliteten av intervjuerna (Bryman & Bell, 2015). Efter transkriberingen sammanställde vi de svar som var relevanta för vår undersökning i tabellerså läsaren enkelt kan ta del av dessa. Vidare genomförde vi sedan en ytlig analys av de svar vi fått in från de olika respondenterna. Detta förde oss sedan in på diskussionskapitlet där vi diskuterade de olika teknologiska förutsättningarna som de olika aktörerna hade och hur vi kunde se en koppling mellan detta och deras hantering av teknisk skuld.

3.5 Etik

Etiken behandlar de etiska överväganden som forskare som studerar människor och människors olika sammanhang tvingas göra. Etiken är central för forskningsvärlden och dess framtida värde (Raune, 2006). Janet M. Raune (2006) talar om 5 olika etiska regler vid kvalitativ

forskningsmetod. Dessa regler är; Undersökningen ska inte medföra några skadliga konsekvenser för deltagarna, deltagarna ska ge sitt samtycke till att delta, forskarna ska respektera deltagarnas privatliv, forskarna ska sträva efter att undvika intressekonflikter och Etisk rapportering - hela sanningen ska sägas, inget annat är sanningen (Raune, 2006). Dessa aspekter har vi haft i åtanke vid utformandet av frågor och särskilt i genomförandet av intervjuerna.

Den första principen, *Undersökningen ska inte medföra några skadliga konsekvenser för deltagarna*, behandlar forskningsaktiviteter som kan skada eller innebära oskäliga risker för deltagarna. Exempel på sådana skador som tas upp i Raune (2006) är deltagarnas fysiska, psykiska och emotionella välbefinnande. Vår bedömning är att deltagarna inte löper någon risk att skadas av undersökningen, även om det som Raune (2006) argumenterar för kan vara svårt att i förväg känna till eventuella negativa konsekvenser en studie kan ha.

Under *samtyckesprincipen* finns en del aspekter såsom **Kompetens**, vilken syftar till respondentens förmåga att själv avgöra huruvida den vill delta i studien. Eftersom samtliga i undersökningen utan tvivel har ansvarsförmåga att själva göra denna bedömning har vi frågat respondenten om samtycke, vilket i samtliga fall beviljats. En annan aspekt är **utförlig information** - Denna aspekt av samtycket förutsätter att deltagaren tilldelats tillräcklig och relevant information om studien för att fatta beslut om huruvida de vill delta i studien eller ej. Därför har vi innan intervjuerna skickat en mall med de ämnen vi tänkt fråga kring samt exempel på frågor som kommer ställas under intervjun. **Fattningsförmåga** - Beskriver att samtliga respondenter måste förstå den information de delgivits innan intervjun, vilket vi försöker säkerställa genom att fråga ifall några frågor är oklara (Raune, 2006).

Nästa princip - *Forskarna ska respektera respondentens privatliv*, innefattar olika aspekter som syftar till att skydda respondentens privata uppgifter. De privata uppgifter som hanteras i denna studie är respondentens namn och i vissa fall respondentens roll på företaget. Därför frågar vi innan varje intervju ifall respondenter önskar vara anonym. Således är den enda aspekt kring personliga uppgifter som blir applicerbar för oss **spridning av forskningsresultat**. Denna aspekt behandlar anonymitet och offentliggörande av personlig information. Eftersom denna uppsats är tänkt att publiceras måste löftet om anonymitet mot de respondenter som önskat vara anonyma uppfyllas genom att organisera insamlad data på ett sätt att inga kopplingar från vad som sagts i intervjun kan göras till individen i fråga (Raune, 2006).

Den fjärde principen säger att *forskarna ska sträva efter att undvika intressekonflikter*. Forskare i allmänhet är mycket engagerade i en objektiv och opartisk insamling av information. Det är emellertid inte ovanligt att näringslivet och forskningen har kopplingar som kan väcka frågor kring validitet och reliabilitet i en forskningsstudie, särskilt när exempelvis företag anslår medel för forskning (Raune, 2006). I denna studie undersöker vi hur olika aktörer, med olika förutsättningar, som är verksamma i samma bransch arbetar med framförallt teknisk skuld samt diskuterar och ställer dessa mot varandra i syfte att besvara vår frågeställning. Vi har därför genom hela arbetet beaktat den egna objektiviteten och försökt bevara den kritiska synvinkeln som vi gått in i arbetet med.

Den sista principen som tas upp i Raune (2006) är *etisk rapportering - hela sanningen och inget annat än sanningen*. Empirisk forskning utgör en trovärdig kunskapskälla. Därför måste forskare

vara uppmärksamma på vilken betydelse deras rapporter har för spridandet av kunskap och sanning. Forskningsresultaten och statistik i rapporter är ett resultat av beslut som tagits av forskaren om bland annat mätning, urval, undersökningsdesign och analys av resultaten. Dessa resultat ligger ibland som grund för olika argument, inte minst politiska. Forskare bör därför på ett vördnadsfullt sätt fatta beslut om vilka resultat som ska visas. Att ta fram en särskilt resultat och endast beskriva viss statistik istället för annan kan innebära två helt olika berättelser som har sin grund i samma data (Raune, 2006). Vi vill därför ge en rättvis och beskrivande bild av det område vi undersöker och försöker framställa statistik och data, i det omfång vi gjort detta, på ett korrekt och icke missvisande sätt.

3.6 Validitet och Reliabilitet

3.6.1 Validitet

Jacobsen (2002) förklarar validitet som intersubjektivitet, vilket är när flera personer har samma uppfattning. Vidare delar Jacobsen (2002) upp validitet i intern- och extern validitet.

Våra intervjuobjekt är väldigt relevanta för studien då de har flera års erfarenheter inom IT och besitter centrala IT-positioner hos banker i Sverige. Vidare är samtliga intervjuobjekt väl bekanta med begreppet teknisk skuld. Enligt Jacobsen (2002) uppfyller man intern validitet om man som forskare kan vara säker på de slutsatser man drar stämmer (Jacobsen, 2002), vilket vi uppfyller genom det tidigare nämnda.

För att uppfylla extern validitet, alltså att resultatet är generaliserbart (Jacobsen, 2002), intervjuade vi personer som jobbade på banker med olika teknologiska förutsättningar då de vuxit fram under olika tidsepoker. Således blir studien relevant för andra marknader som innehåller traditionella aktörer samt andra aktörer som vuxit fram på senare tid tack vare digitaliseringen, och då med helt andra teknologiska förutsättningar.

3.6.2 Reliabilitet

Enligt Jacobsen (2002) så betyder reliabilitet att data som samlas in måste vara pålitlig. Därför är det väldigt viktigt att man värdesätter hur intervjuens upplägg ser ut för att det inte ska påverka resultatet. Vi som intervjuare var således noggranna med att tydliggöra att informationen som delas under intervjun endast är till för rapportens syfte och inte kommer användas för annat bruk. Genom att även tydliggöra att intervjupersonen och dess organisation kunde vara anonyma, och att denne får ta del av den färdigställda rapporten, kunde vi få intervjupersonen att känna sig lugn och inte pressad eller nervös att säga fel. Sådan press och nervositet kan annars leda till opålitliga data enligt något som Jacobsen (2002) beskriver som undersökareffekten. Vi skickade även ut intervjumallen till respondenterna 1–2 dagar innan intervjun för att dessa skulle kunna förbereda sig och ge så korrekta och utförliga svar som möjligt. På så vis höjdes även reliabiliteten av intervjuerna.

4 Empiri

I kommande kapitel presenteras det empiriska resultatet från våra fem intervjuer. I tabellerna redovisas samlade svar från de olika respondenterna som behandlar respektive undersökningsområde. Anledningen till detta är att intervjuerna var semistrukturerade och svaren på frågorna därför har framkommit i olika delar av respektive intervju. Fullständiga intervjuer återfinns i appendix.

4.1 Förekomst av teknisk skuld

Respondenterna blev tillfrågade ifall teknisk skuld förekom hos verksamheten. Samtliga respondenter svarade att de var väl bekanta med uttrycket och att deras verksamhet i olika hänseenden var påverkade av den. Förekomsten av skuld beskrevs ofta uppkomma i tekniska miljöer som rörde koden. Även annan skuld togs upp, exempelvis skuld som uppstår till följd av regelverk, infrastruktur och konkurrens, men denna skuld uppkom i slutändan i systemen. Denna del behandlar framförallt fråga 7, men vi har även kopplat relevanta delar av intervjuerna som besvarar denna fråga då besvarandet uppkommit senare i intervjun.

Respondent	Förekomst av teknisk skuld i verksamheten
Respondent 1	<i>“Alla teknikbolag har legacy. Om de inte startade igår. Men så är det ju. Vi har teknisk skuld och vi har legacy, sen har vi ganska bra koll på den. Men vi kan som alla andra bli bättre på att ta hand om den [...]” Appendix 1:17</i>
Respondent 2	<i>“Ja, alltså jag vill hävda att alla verksamheter som skulle säga något annat skulle ljuga[...] ” (på frågan om de har någon teknisk skuld i verksamheten) “[...] Så det är klart att det kommer finnas kompromisser, med saker som måste ut på marknaden här och nu, och problem som man får återkomma till och förbättra lite senare. Det är en naturlig avvägning för oss liksom. Så skulden finns där och den hanteras absolut men går inte att undvika till fullo.” Appendix 2:10</i>
Respondent 3	<i>“Absolut, det är en central del av vårt arbete så det känner jag till väl.” Appendix 3:14</i> <i>“Ja dels är det ju äldre system, så det kan tex vara mycket småfixar som kanske något temporärt krav som blivit permanent. Det är mycket gamla lösningar och gammal arkitektur [...].” Appendix 3:16</i>
Respondent 4	<i>“[...]det där har vuxit över åren så att nu är liksom i formatet helt omöjligt att hantera om man ska få allting att funka STP, alltså straight through processing”. Appendix 4:15</i>

Respondent 5	<p><i>“[...]och det går ju end of life om ett par år så att vi måste se till att har vi några system som kommer bli end of life så småningom då måste vi börja i god tid för att byta ut dem eller uppgradera dem för det är ju en tröghet i affärssystem och så vidare att byta ut såna där saker”. Appendix 5:16</i></p> <p><i>“[...]Sen kan man ju säga att det har kanske med skuld att göra lite grann eftersom vi har ju byggt in oss lite grann i det nuvarande systemet då men så det är en beroende på definitionsfråga hur man ser det.” Appendix 5:18</i></p>
--------------	--

Tabell 5: Förekomst av teknisk skuld

Samtliga respondenter beskriver att de på olika sätt kan se teknisk skuld i verksamheten. Eftersom de tillfrågade har olika positioner på företagen är det möjligt att synen på skulden är påverkad av detta och därför beskriver de skuld i olika egenskaper. Respondent 1 och 2 beskriver båda att de tror att alla företag med tekniska artefakter har teknisk skuld. Vidare beskriver också respondent 1 att det alltid finns möjlighet att hantera skulden på ett bättre sätt. Vi kan också se att respondent 1, 3 och 4 påpekar att tiden systemen varit i bruk leder till en teknisk skuld. Respondent 5 antyder också på detta då han menar att uppgraderingen av systemen är av betydelse för mängden skuld som uppstår.

4.2 Teknisk skuld och system

Respondenterna blev i fråga 16a tillfrågade om de ansåg att åldern på systemen påverkade mängden teknisk skuld samt hanteringen av denna. Det framgick i samtliga intervjuer att de tillfrågade ansåg att åldern på systemen, och i synnerhet mängden legacy i systemet, var av stor vikt för mängden teknisk skuld.

Respondent	Teknisk skuld och system
Respondent 1	<p><i>“Vår senaste version av arkitektur tillåter deployment av applikation precis när som helst oberoende av alla andra applikationer och vissa applikationer är inte byggda på det sättet ännu. På så sätt lever våra system kvar i ett lite äldre releaseprocess till exempel på grund av tekniska lösningar som finns i botten”. Appendix 1:20</i></p> <p><i>Följdfråga: “Jag måste bara fråga då jag blir intresserad. När du säger det där med att legacy finns i alla system, skulle du då påstå att det finns i olika grad? Jag tänker på just företag som utvecklats mycket tidigare än er med betydligt äldre system”. Appendix 1:21 & 1:23</i></p> <p><i>“Ja precis, då kan man ju sitta med legacy i form utav teknikstackar som inte ens underhålls längre. Det är inte läget på avanza, vi kör ju java rakt igenom och det vi kallar gammalt är liksom 10 år gammalt så det är ofta rätt</i></p>

	<p>välfungerande. Sen har vi några komponenter som är 15 år gamla, men de är inga problem att underhålla”. Appendix 1:24</p> <p>“[...] skulle Java fortsätta sin utveckling som de gjorde fram till version 8, då det gick fruktansvärt långsamt under många, då hade jag varit mer orolig än vad jag är nu när de faktiskt fått fart på språket igen.” Appendix 1:56</p>
Respondent 2	<p>“Ja men det är väl klart att det finns områden som kanske varit med från start, som har varit med längre. De skulle kunna ha ett större element av det där. Och där man i ett tidigare skede kanske hade andra ingångsvärden om vad som värdesätts. Man kanske fokuserar på en typ av produkt eller en typ av marknad och sen när man har fler produkter och går ut mot fler marknader då kan det ju finnas skuld relaterat till det.” 2:14</p>
Respondent 3	<p>“Ja dels är det ju äldre system, så det kan tex vara mycket småfixar som kanske något temporärt krav som blivit permanent. Det är mycket gamla lösningar och gammal arkitektur. Det är ju system som liksom byggts på varandra, man kanske börjar med en fix här och där och sen kanske det kommer något slags lagkrav som gör att vi måste komplettera med det här osv. [...]” Appendix 3:16</p> <p>“Ja alltså det är klart att det är positivt att det är ett ungt system på det sättet och verktygen för det och förutsättningarna att arbeta med det blir lättare med en moderna arkitektur. [...]” Appendix 3:42</p>
Respondent 4	<p>“[...] Det vill säga utan manuell påverkan och problemet som vi har är att de här systemen som vi har de är, jag tror ungefär 30 år gamla. De utvecklades i början av av nittioalet ungefär”. Appendix 4:15</p> <p>“[...]Exempel Cobol är liksom inte riktigt kompatibelt med variabla fältlängder och dynamiska strukturer. Vad vi gör då är helt enkelt att vi gör ett och det är nog ganska typiskt för hur man löser problemet. Att vi har ett java skal kan man säga runt omkring våra gamla legacysystem”. Appendix 4:15</p> <p>“Ja, men det är så här att problemet med att bygga helt nytt är att det finns... alla de här gamla systemen innehåller liksom verksamhetsregler som har vuxit fast och vuxit fram under kanske 30 års tid och vi hanterar ju dels *storbank* sen har vi alla enheter runt om i världen, alltså till Tyskland, Norge, Finland och så vidare [...]”. Appendix 4:19</p>

Respondent 5	<p><i>“[...]men sen finns det ju den andra delen som vi var inne på med kärnsystem som man har haft väldigt länge och som man har byggt på och byggt på. Återigen Carnegie är ju ganska ungt bolag så vi har ju inte funnits i såg 50 år utan våra system är hyfsat moderna om man kallar det för det. Så att vi tittar ju på det där men ur vår aspekt så är den strategi som vi har när det gäller digitalisering kräver att vi byter ut en hel del av vår grundprogramvara så att säga, våra kärnsystem”. Appendix 5:16</i></p> <p><i>“Vi har inte några stordatorer som står liksom och tuggar i någon källare sen 20 år tillbaka, vi är för unga för det så att säga rent systemmässigt. Så vi kanske inte har den skulden på det sättet att vi helt plötsligt nått vår livs ände utan det handlar mer om strategier att vi för att kunna ta klivet fullt ut mot digitalisering så behöver vi ju förändra vår grundstruktur.” Appendix 5:20</i></p>
--------------	--

Tabell 6: Teknisk skuld och system

Samtliga respondenter ansåg att ålder på systemen på något sätt påverkade mängden teknisk skuld. Respondent 3 och 4, som båda upplever en relativt hög grad legacy i de systemen de arbetar, menar att de på grund av detta har en större teknisk skuld och ett större arbete med uppdateringar och förändringar. Respondent 4 beskriver att anpassningen av legacysystemen till en nyare arkitektur och förändrad infrastruktur är ett exempel på skuld som uppkommer vid hantering av äldre system. Samtidigt beskriver han vikten av att dessa system bevaras då de innehåller verksamhetsregler som arbetats fram under 30 års tid. Respondent 3 beskriver liknande problematik med äldre arkitektur och äldre lösningar, som utgör kärnfunktioner för verksamheten men som måste kunna kommunicera med nyare system. Respondent 3 förklarar också att en stor del av dennes arbete går ut på att utveckla denna kommunikation. Både respondent 3 och 4 talar om system i lager och beskriver hur den nya arkitekturen byggts ovanpå eller “omkring” den äldre. Övriga respondenter ser också problematiken men uttrycker den inte lika tydligt eller på samma sätt. Respondent 2 och 5 menar att de inte har särskilt mycket legacy och därför inte upplever så mycket problematik, medan respondent 1 säger att viss legacy finns, men är inte så omfattande. Dessutom försöker de uppdatera den ständigt för att minska problematiken. Respondent 1 påpekar också att teknisk skuld minskar då de genomgripande jobbar med ett och samma programmeringsspråk.

4.3 Teknisk skuld och arkitektur

Respondenterna blev tillfrågade att övergripligt beskriva arkitekturen för företagets olika system och om de tror att val av arkitektur har någon påverkan för mängden teknisk skuld som uppstår. Vidare fick de frågan om, och i så fall hur, arkitekturen i sin tur påverkar hantering av skulden. (Frågor: 14 & 19–21)

Respondent	Teknisk skuld och arkitektur
------------	------------------------------

Respondent 1	<p><i>“Där har vi en mikrotjänstarkitektur som vi byggt på sedan 2011. Vi började bygga mikrotjänster innan begreppet fanns då det myntades först 2012.” Appendix 1:44</i></p> <p><i>“Det är så otroligt viktigt. Asså det vi inte visste inom Avanza när vi höll på att bygga det 2007 var ju hur vi bygger ett system som vi om 3 år kan förändra”. Appendix 1:62</i></p> <p>Följdfråga: “Hur menar du att man bygger ett system som är lätt att hantera 3 år senare?” Appendix 1:63</p> <p><i>“Ja men asså mikrotjänstarkitekturen är ju svar på det. För då minskar ju scopet i dina applikationer och mindre applikationer är ju lättare att förändra, dem är ju lättare att bygga om. En annan viktig sak med att bygga mikrotjänstarkitektur är att du ska ju gömma data, du ska gömma den interna datarepresentationen mellan mikrotjänsterna så mikrotjänst A ska ju inte veta något om hur mikrotjänst B hanterar informationen. Och det gör ju att du kan ändra mikrotjänst B, för det finns inget beroende när du delar din databas exempelvis som du gör med gamla system då är det jättesvårt att ändra datan”. Appendix 1:64</i></p> <p>Följdfråga: “Det är väl en av de stora fördelarna med mikrotjänstarkitektur att det är så pass flexibelt gentemot andra arkitekturer?” Appendix 1:65</p> <p><i>“Absolut, flexibilitet och ja det blir ju i flera dimensioner. En dimension är ju att ändra saker och en annan är just skalning. Vissa av våra system måste vi skala upp i 130 instanser och då kan vi ju göra det medans andra inte behöver vara så uppskalade. Den problematiken är ju mycket enklare än att försöka sig på en jättemonolit och se till att den fungerar bra”. Appendix 1:66</i></p> <p>Följdfråga: Ser ni några nackdelar med användandet av mikrotjänster? Appendix 1:67</p> <p><i>“Jo men det är klart det gör det, man stöter på rätt mycket det är ingenting man börjar med och kör. Vi har ju varit med på resan länge så vi har ju löst många av våra problem. Det första man stöter på var manageringen av de här grejerna. Avanza, vi har 1400 mikrotjänster och 6000 instanser, det är liksom ingenting man bara gör. Det här ska ju funka jämt, så hur lyckas man deploya en sån miljö varje vecka eller till och med oftare, bara den frågan är ju svår. Hur gör man med övervakning av de här systemen? Hur gör man med feltolerans? Det är ju en superviktig fråga”. “[...]Har du så många program som vi har då kommer det gå sönder och det går sönder saker jämt och det måste du hantera och det ger ju en lite mer komplex programmeringsmodell. Så det är ju sånt man stöter på, och vad blir det mer.. det är klart det blir ju andra typer av utmaningar vid deployment och så vidare. Det gäller ju att</i></p>
--------------	---

	<i>bygga tjänster så att de inte har för hårda kopplingar till varandra.” Appendix 1:68</i>
Respondent 2	<p><i>“[...]det är positivt att det är ett ungt system på det sättet och verktygen för det och förutsättningarna att arbeta med det (=teknisk skuld) blir lättare med en modern arkitektur. Så för vår del är det en väldigt positiv sak.” Appendix 2:42</i></p> <p><i>“Kan du se en koppling mellan mikrotjänst-arkitekturen och hanteringen av teknisk skuld? Att den eventuellt är lättare att hantera i och med arkitekturen?” Appendix 2:50</i></p> <p><i>“Ja jag skulle säga att det är både och lite beroende på hur man ser det men jag skulle ändå säga övervägande att det är bättre för vår del. Sen kan det såklart bli komplext när det blir väldigt många services med beroenden dem emellan och så att ha en komplett översikt av alla delar och var skulden finns. Det går att argumentera för det i alla fall.” Appendix 2:51</i></p>
Respondent 3	<i>“Absolut. Dels är det ju att det finns så extremt många samband mellan de olika systemen, och dels är det ju att det är äldre system och det är ingen hemlighet att man inte har samma möjligheter att jobba med DevOps och jobba med continuous integration. Enhetstester kanske inte heller finns på samma sätt och det tar längre tid att utveckla saker och skriva om saker, och man kan inte automatisera testandet på samma sätt.” Appendix 3:18</i>
Respondent 4	<i>“[...] Vad vi gör då är helt enkelt att vi gör ett och det är nog ganska typiskt för hur man löser problemet. Att vi har ett java skal kan man säga runt omkring våra gamla legacysystem. Om vi säger att vi tar en ankommande betalning som kommer så.. Består den kanske av totalt sett 700 fält 685 av de där är rent trams. Det är ingenting som är vi är intresserade av det går att ange allt ifrån, liksom skonumret på den som skickar betalningen till liksom var vederbörande är född någonstans och vilket land och vilket datum och någon gång kommer säkert att användas. Men men det vi är intresserade av primärt, det är de 15 fälten som beskriver en betalning idag. Därför det är vad våra legacysystem hanterar. Vad vi gör då är att vi bygger en översättare som är ligger i vår moderna utvecklingsmiljö, som är java-driven. Den ligger som ett stort block framför som plockar in den där XML:en, hämtar ut de fält som är av intresse och sparar ner resten i en databas. Sparar ned vad vi fick in sparar ner också vad vi skickar ut till legacy-systemen och sen skickar vi betalningen i ett slags hybrid format där vi har plockat ut de XML-taggar som är liksom är intressanta för det mottagande systemet och det blir ju som en Cobol-koppling, och allt med allt vad det innebär i Cobol. Och det här innebär att om vi har en betalning då som exempelvis ska vända hos oss den ska liksom in i banken och ska nu ut igen därför att den skulle inte stanna hos oss. Vi är bara en intermediär. Då kan vi fiska upp de fälten som vi så att säga</i>

	<p><i>har filtrerat bort och slängt på vägen och se till att de går iväg i den avgående betalning som går ut igen. Och det är liksom det där gör ju det att utifrån sett så ser det ut som att vi hanterar alla de här sjuhundra taggarna, men egentligen gör vi inte det. Vi hanterar fortfarande de 15, 16, 17, 18 som vi liksom är intresserad av. Och det här systemet då, som som jag jobbar med idag, det har 5 eller 6 olika, liksom utgångar inklusive Swift gateway då som förbinder liksom banken med Swift, alltså stora betalnätet. Och sen får vi betalningar, dels från några hundra banker runt om i världen plus 2 stora clearinghus som hanterar euro och det där. Det där är ju ett sätt att förlänga livslängden hos våra cobolssystem med med 5 till 10 år kanske”.</i> Appendix 4:15</p> <p>“Det här med mikrotjänstarkitektur har blivit ganska centralt efter PSD2 och open banking i allmänhet, är det någonting du ser eller vad är din tanke om det?” Appendix 4:50</p> <p><i>“Jag är fel person att svara på frågan för att jag är inte arkitekt, utan jag är kravare och även om jag tänker som en arkitekt så är det inte riktigt mitt arbetsområde att fundera över den typen av grejer. Men det är klart att det pratas om mikrotjänster i alla banker och har blivit alltmer centralt”.</i> Appendix 4:51</p>
Respondent 5	<p><i>“Så att dit vi är på väg är att vi håller på att sticka upp det här med mikrotjänster. Vi använder ju den typen av teknik också, så att vi kommer ju att splitta upp hela våran, så att säga grundstruktur och bygga små byggklossar istället, små mikrotjänster. Så vi kommer ju bygga hela våran nya IT infrastruktur på en ny teknik”.</i> Appendix 5:34</p> <p>“Och hur ser du på det bytet av arkitektur till MSA (Microservice architecture)?” Appendix 5:37</p> <p><i>“Det ställer ju utmaningar på hela organisationen, i synnerhet för mig och IT och även hela projektorganisationen också. Vi kommer ju använda en ny teknik, så det ställer ju även krav på både mjukvara och teknik och sådana saker, och det kanske då även öppnar upp för nya tjänster inom Microsoft. Vi rör ju oss långsamt mot molnet så att säga och där finns ju många Cloud tjänster som vi också är intresserad av att titta på eftersom Cloudtjänster har många fördelar. Det går ju snabbare att implementera och så vidare. Det finns en större kompetens runt där så att vi rör oss ju också åt det hållet och det ställer ju utmaningar för mig runt säker som utveckling, att vi har de verktyg vi använder när vi bygger och när vi kontrollerar våra tjänster eller byggen att de kan se liksom det nya vi håller på med”.</i> Appendix 5:38</p>

Tabell 7: Teknisk skuld och arkitektur

4.4 Hantering av teknisk skuld

Respondenterna fick i frågorna 10 & 13 svara på frågor om hur respektive organisation mer ingående arbetade med hanteringen av teknisk skuld och ifall de använde några särskilda strategier, som de vi ser i Li. et al (2015).

Respondent	Hanteringen av teknisk skuld
Respondent 1	<p>“[...] Vi sätter liksom inte någon prislapp på grejerna men generellt sätt gör man det klassiska diagrammet med kostnad att åtgärda på ena axeln och kostnad att ha kvar den på den andra axeln och sen ser man vilka skulder som bör åtgärdas. “Appendix 1:32</p> <p>Följdfråga: “Så en typ av prioriteringslista?” Appendix 1:33</p> <p>“Ja det blir ju det. Det som faller ut från det diagrammet blir en typ av prioritering. “Appendix 1:34</p>
Respondent 2	<p>“Ja alltså dels är det ju det här. Vi jobbar ju agilt, vi jobbar i sprinter med 2 veckors cykler med planeringar och standups och allt det här som ni säkert är bekanta med, och där kommer ju det här in också som en del av det. Det finns ju alltid en grej med att få in maintenance som en del av sprintarna också och vi har även haft rena sprintar som inte fokuserat på nya features utan bara ta hand om eventuell skuld som har funnits. Så det är väl den approach vi har. Vi delar i backloggen som alla andra saker som sedan hanteras av våra team och utvecklare”. Appendix 2:18</p> <p>“Skulle du säga att det finns något specifikt sätt ni mäter eller utgår ifrån när ni kollar på teknisk skuld? Kollar ni till exempel på kostnad mot effekt av att återbetala skulden?” Appendix 2:21</p> <p>“Ja de tankarna finns ju alltid där. Cost-benefit-delen är ju central i allt så att säga. Särskilt ifall man har kanske begränsad kapacitet i ett mindre bolag och olika typer av fokus på vad som är viktigast för stunden”. Appendix 2:22</p>
Respondent 3	<p>“Jaddå, en strategi är tex att så fort man ska in och ändra i något gammalt så ska man också gå igenom det och se till så att det inte finns någon gammal eller förlegad kod. Man ska alltså fixa till allt som rimligt kan förbättras. Men det är oftast inget aktivt arbete kring det som till exempel att vi skulle ha särskilda sprints ägnade åt att enbart hantera teknisk skuld. Istället blir det mer inbakat i att om vi ändå ska in och ändra i det här programmet så ser vi till att försöka förbättra det också.” Appendix 3:20</p>
Respondent 4	<p>“[...]Vad vi gör då är att vi bygger en översättare som ligger i vår moderna utvecklingsmiljö, som är java-driven. Den ligger som ett stort block framför</p>

	<i>som plockar in den där XML:en, hämtar ut de fält som är av intresse och sparar ner resten i en databas. Sparar ned vad vi fick in sparar ner också vad vi skickar ut till legacy-systemen och sen skickar vi betalningen i ett slags hybrid format där vi har plockat ut de XML-taggar som är liksom är intressanta för det mottagande systemet och det blir ju som en Cobol-koppling, och allt med allt vad det innebär i Cobol [...]” Appendix 4:15</i>
Respondent 5	<i>“Jo men man har ju alltid en dialog med verksamheten. Om man har ett system som funkar så kan man ibland känna att varför ska jag som systemägare eller som ansvarig för en del av verksamheten lägga pengar på att uppgradera ett system som redan fungerar och innehåller de möjligheter och funktioner som jag vill ha. Ibland så måste man då ha en lite mera tydlig dialog med verksamheten tala om att för att vi inte ska hamna efter, alltså ålägga oss med teknisk skuld eller vad det kan va för någonting, eller att de blir out of date och inte supporterade längre, så måste du lägga pengar på att uppgradera. Du kanske inte får så mycket funktionalitet eller det kanske inte blir så stor nytta för dig.” Appendix 5:30</i>

Tabell 8: Hantering av teknisk skuld

Sammantaget kan vi se att samtliga aktörer arbetar med att förhindra eller hantera teknisk skuld på något sätt. Respondent 1 beskriver att de arbetar med ett cost-benefit diagram och någon form av prioriteringsordning. Det samma gäller respondent 2 som beskriver att cost-benefit delen finns med övergripande i alla delar, detta beskrivs dock först efter vi frågat om specifikt cost-benefit är en del av det. Respondent 3, 4 och 5 beskriver också alla olika hantering av skulden så som respondent 3: “Vi ser över det så fort vi ska in i något gammalt”, eller som respondent 5 som arbetar mycket med att förhindra skulden redan innan den uppstår.

5 Analys & Diskussion

5.1 Strukturomvandling

Något som gemensamt framkommit under både vår teorigenomgång och empiriska insamling är att de olika aktörerna har haft olika teknologiska förutsättningar under den digitala strukturomvandling som finanssektorn genomgått. Precis som Svenska Bankföreningen (2019) beskriver har de traditionella bankaktörernas affär funnits långt innan digitaliseringen uppkom. Detta kan ställas mot många av de modernare aktörerna, som till exempel Avanza och Rocker, vilka istället har uppkommit som en direkt följd av digitaliseringens möjligheter (Appendix 1:6 & Appendix 1:4). Dessutom består de traditionella bankaktörernas verksamhet av flera delar som i mångt och mycket innefattar hela den traditionella värdekedjan inom bankväsendet, medan de modernare aktörerna istället brukar nischa sin verksamhet till en särskild del av värdekedjan. Denna fördelning har vidare drivits på med hjälp av PSD2, vilket har öppnat upp för fler aktörer på en marknad som tidigare dominerats av ett mindre antal aktörer. Detta framkommer även i våra intervjuer där bland annat respondent 1, 2 och 3 delar synen av att PSD2, och Open banking i allmänhet, möjliggjort för mer teknikdrivna aktörer att utveckla integrerade tjänster på ett enklare sätt (Appendix 1:8, Appendix 3:4 & Appendix 3.10). I intervjun med respondent 3, en systemutvecklare på en av storbankerna, beskriver hen bland annat hur konkurrensen på marknaden öppnats upp då nyare aktörer kan slå sig in och fokusera på de mest lukrativa verksamhetsdelarna och dessutom dra nytta av den infrastruktur som bankerna redan byggt upp (Appendix 3:10). Detta underbyggs både i vår litteraturgenomgång och i intervjuerna, där till exempel respondent 1, CTO på Avanza, menar att mindre uppstickare ofta fokuserar på en specifik sak som de gör väldigt bra, vilket bankerna kan ha svårt att konkurrera mot (Appendix 1:74).

5.2 Teknisk skuld & förlegade system

Vi ser skillnader mellan de olika aktörerna gällande åldern av deras system och hur deras kärnverksamhet såg ut initialt. Storbankernas affär har traditionellt sett drivits från bankkontor med penna och checkhäfte (Appendix 3:28), medan affären hos de digitala bankerna istället har grundat sig i att göra bankvärlden digital. Med andra ord utvecklades systemen hos de digitala bankerna med fundamenten att förenkla diverse områden av banktjänster för slutkunden, medan grundtanken med den initiala systemutvecklingen hos de traditionella aktörerna var att stödja verksamheten som den såg ut då. Detta understryker även respondent 3 som menar att de system som initialt utvecklades hos storbankerna, utvecklades som stöd för den tidens verksamhet och inte för slutkunden (Appendix 3:30). Dessa system har sedan successivt byggts på för att anpassas efter hur marknaden utvecklats och även de förändrade lagkrav som tillkommit under tiden (Appendix 3:16). Respondent 3 påpekar bland annat att stora delar av deras IT-arkitektur är byggd i början på 90-talet (Appendix 3:30), vilket även respondent 4 antydde på (Appendix 4:15). Både Respondent 3 och 4 berättar vidare att organisationerna fortfarande arbetar med programmeringsspråket Cobol (Appendix 4:15 & Appendix 3:2). Cobol är ett tredje generationens programmeringsspråk och i linje med vad Bennet (1994) menar att förlegade system ofta är skrivna i. Respondent 3 berättar också att systemdokumentationen ofta kan vara svår att gå igenom då många av systemen funnits i decennier samtidigt som dokumentationen

förändrats flertalet gånger genom årens gång (Appendix 3:36). Även detta är i linje med vad både Warren (1999) och Bennet (1994) menar tyder på ett förlegat system. Ställer man detta i korrelation med vad övriga respondenter antydde på så kan man förstå att deras teknologiska förutsättningar skiljer sig åt. Som tidigare nämnt tillkom de digitala bankernas, i det här fallet Avanza och Rocker, organisationer tack vare digitaliseringen. Därför är det naturligt att deras grundsystem är yngre. Respondent 1 menar till exempel att det de kallar gammalt gällande deras systemarkitektur är cirka 10 år gammalt (Appendix 1:48). Dessutom består deras grundsystem genomgripande av Java (Appendix 1:24), vilket kan ställas mot mer traditionella aktörer vars kärnsystem ofta består av Cobol eller annat tredje generationens språk. Vidare antydde respondent 2 att deras system inte är äldre än 4-5 år (Appendix 2:37), vilket är naturligt då Rocker startades 2017, och respondent 5, CIFO på Carnegie, berättar att de inte heller besitter särskilt gamla system (Appendix 5:44). Carnegies verksamhet skiljer sig å andra sidan åt mot övriga nämnda banker, men det är inget som förändrar något i just det här avseendet. Således är det tydligt att de olika aktörernas teknologiska förutsättningar skiljer sig åt.

Dessa teknologiska förutsättningar blir av särskild betydelse då vi kommer in på teknisk skuld, och då framförallt på mängden teknisk skuld. De organisationer som besitter system med en högre grad legacy, tycks också ha mer underhållningsarbete och i allmänhet mer teknisk skuld. Detta beskrivs av Fowler (2019), Allman (2012), Suryanarayana et al. (2015) och återges även i intervjuerna, där till exempel respondent 3 och 4 gemensamt påpekar att äldre system är svåra att underhålla och kräver mycket extra arbete på grund av gamla lösningar och gammal arkitektur (Appendix; 3:16, och 4:15). Således blir kopplingen mellan förlegade system och förekomsten av teknisk skuld väldigt tydlig.

5.3 Teknisk skuld & arkitektur

En annan förutsättning som tycks vara av stor betydelse för skulden är val av arkitektur. Detta har så klart också starka kopplingar till åldern på systemen och mängden legacy i dem. Warren (1999) beskriver hur gammal arkitektur kan försvåra för underhållning och förändringar av datastrukturen och som framkommer i vår teori är mikrotjänstarkitektur ett sätt att underlätta denna svåra hantering. Detta framkommer även i intervjuerna där vi finner ett ökat intresse för just mikrotjänst-arkitektur, vilket även kan bero på att denna arkitektur underlättar för gränssnittskommunikation genom APIer, vilket Open Banking bygger på. Vi ser vidare att de som redan använder denna arkitektur upplever att den underlättar för hanteringen av teknisk skuld. Exempelvis menar respondent 1 på Avanza att äganderelationen mellan utvecklingsteamet och mikrotjänsterna är en nyckel till en god underhållning av systemen (Appendix 1:62, 1:64). Respondent 2 på Rocker menar också att användandet av denna arkitektur underlättar för hantering av skulden (Appendix 2:42). Respondent 5 på Carnegie beskriver att de befinner sig i en ombyggnadsfas där de nu implementerar en ny mikrotjänst-arkitektur. Detta menar han dock även beror på andra fördelar såsom molntjänster, snabbare lanseringar och större kompetens (Appendix 5:38). Som tidigare nämnt arbetar respondent 1, 2 och 5 med liknande arkitekturer. Respondent 3 vill inte kommentera för mycket kring mikrotjänster men känner igen att den nuvarande arkitekturen i vissa fall kan försvåra för hanteringen av skuld. Bland annat nämner han att det finns många olika samband mellan systemen och beskriver svårigheter att arbeta med devops och kontinuerlig integration (Appendix 3:18). Respondent 4 nämner problematiken för dennes organisation att implementera

nyare system eftersom de delarna av systemen som är äldre behandlar den absoluta kärnverksamheten och innefattar verksamhetsregler som tagits fram under 30 år. Därför blir en omstrukturering i ett sådant fall mycket invecklad och kostsam (Appendix 4:19).

5.4 Hantering av teknisk skuld

Gällande hantering av teknisk skuld insåg vi att inga av de tillfrågade företagen använder sig punktligt av metoderna som tagits upp i teorin. Liknande metoder användes i delar av hanteringen hos enstaka företag men genomgripande ser vi lite korrelation mellan hanteringen hos företagen och teorierna. Resultatet från intervjuerna skilde sig således från teorin. Nedan gör vi en jämförelse av strategier för hantering av teknisk skuld som finns beskrivet i teorin och jämför dessa med vad vi fann i intervjuerna.

Identifiering gjordes inte enligt någon strategi som teorin förordar. I teorin beskrivs identifiering av Li et al. (2015) som en samling aktiviteter som görs för att identifiera karaktär på skulden. Detta kan göras till exempel med workshops där koden analyseras. Respondent 2 från Rocker beskriver hur de ibland har sprintar som endast fokuserar på att hantera en skuld samt att de delar dessa typer av problem i backloggen tillsammans med övriga “fixes”. Respondent 3 beskriver att de hanterar skulden “varje gång vi går in i något gammalt” men att de inte har några särskilda strategier utöver detta. Fler respondenter beskriver också hur det agila arbetet underlättar hanteringen men vi kan inte se att de använder sig av aktiviteter som fokuserar kring identifiering av skulden.

Mätning och Prioritering gjordes till viss del av de tillfrågade företagen. Respondent 1 på Avanza beskriver att de gör någon form av cost-benefit-analys av skulden för att sedan prioritera vilka skulder som behöver återbetalas först. Respondent 2 beskriver också att cost-benefit är centralt vid hantering av skulden. I övrigt fann vi inte att något annat företag arbetade med mätning och prioritering av skulder, enligt de principer som teorin föreslår. Detta kan bero på att flera respondenter förklarade att de befinner sig i digitala omstruktureringar, exempelvis på grund av ny arkitektur, nya förutsättningar eller som både respondent 4 och 5 beskriver att de måste anpassa sig till nya lagkrav och deadlines. Detta kan öka tidspressen för utvecklingarna och göra att de således inte har tid att använda sig av specifika strategier för att identifiera, mäta och prioritera teknisk skuld.

Gällande de strategier som tas upp av Alves et al. (2016), *Portfolio Approach*, *Analytical Hierarchy Process* och *Calculation of TD-principals* finner vi att vissa företag arbetar med delar av dessa. *Portfolio Approach*, som av Alves et al. (2016) anses vara det bästa tillvägagångssättet används inte av något av företagen. Det finns likheter hos Avanza och Rocker, men vi vill mena att *Analytical Hierarchy Process (AHP)* bättre beskriver det som de gör. Denna metod går ut på att jämföra olika alternativ med avseende på vissa kriterier och fastställa en övergripande rangordning för varje alternativ. Andra företag arbetar inte formellt efter någon av dessa strategier vilket kan förklaras av det som tidigare nämnts, tidspress från olika påtryckningar från intressenter, nya lagar och konkurrens. Detta är saker som av teorin tas upp som grund för ökad teknisk skuld. I detta hänseende beskriver intervjuerna det som sägs i teorin då vi tydligt ser att påtryckningar från intressenter, konkurrens och framförallt lagkrav skapar större teknisk skuld.

6 Slutsats

Syftet för den här uppsatsen är att undersöka hur olika aktörer inom banksektorn hanterar teknisk skuld för att vidare jämföra hur deras teknologiska förutsättningar påverkar denna hantering. Därför blev forskningsfrågan: *Hur skiljer sig teknisk skuld och hantering av denna åt mellan bankaktörer med olika teknologiska förutsättningar?*

Genom vår studie kan vi konstatera att de olika bankaktörernas teknologiska förutsättningar skiljer sig åt en hel del. Den digitala strukturomvandling som pågått inom branschen har krävt att bankerna blir mer och mer digitala och kan möta sina kunder på icke fysiska plattformar, det vill säga i olika webbapplikationer. Den snabba utvecklingen inom branschen och de ständigt föränderliga omständigheterna ställer högre krav på aktörerna att digitaliseras, samtidigt som lagar- och regelverksförändringar, tillika högre ställda kundförväntningar, driver på branschen ytterligare i samma riktning. De pressande faktorerna för aktörerna att utveckla sin digitala närvaro och kompetens har alltså blivit fler. I takt med att denna teknologiska omvandling sker och marknads aktörer utvecklar nya digitala tjänster och produkter, eller för den delen uppdaterar och gör förändringar i sina befintliga system, kan vi se att komplexiteten av systemen växer och hanteringen av dessa därmed blir svårare. För aktörerna i fråga gäller det således att vara väldigt noggranna i sitt utvecklingsarbete eftersom det i sin tur påverkar förutsättningarna för hanteringen av den tekniska skulden. Vår tes var att mängden teknisk skuld ökade under dessa föränderliga omständigheter, eftersom utvecklare drar på sig mer teknisk skuld ju mer förändringar som bör göras, då de i dessa fall är mer benägna att ta genvägar för att snabbare färdigställa applikationen. Genom vår studie kan vi därför se att den tekniska skulden faktiskt har ökat i högre grad för de företag som har större mängd legacy i sina system och som därmed tvingats göra större och mer fundamentala förändringar. Detta visas i teorin och bekräftas genom våra intervjuer. I och med detta faktum blir den tekniska skuldhanteringen mer omfattande för de företagen med förlegade system och vi kan se att de teknologiska förutsättningarna har en indirekt koppling till mängden teknisk skuld.

Vidare har vi undersökt hur olika arkitekturer påverkar mängden teknisk skuld samt hanteringen av denna. Något vi kan konstatera är att intresset kring mikrotjänster har ökat i branschen under de senaste åren. Detta beror delvis på att det är en snabbare och mer flexibel utvecklingsmiljö, men det är även på grund av den förändrade spelplan som PSD2 och Open Banking har inneburit. Därav är det naturligt att denna typ av arkitektur börjat premieras av de aktörer som försöker utnyttja PSD2 till fullo. Detta ökande intresse tycks vara berättigat eftersom mikrotjänstarkitektur uppenbarligen ger en del fördelar, dels i utvecklandet av nya tjänster och produkter som tycks gå fortare, men i synnerhet för underhållning av systemen i längden. Mikrotjänsternas flexibilitet och låga beroendegrad sinsemellan tycks hjälpa underhållet av systemen eftersom de blir mycket mer ombytliga, vilket vidare tycks förenkla hanteringen av teknisk skuld. Detta är även i linje med vad de företag som använde sig av mikrotjänstarkitektur uttryckte. Vi vill emellertid understryka att mikrotjänstarkitektur är betydligt svårare att införa hos de företag vars största utmaning ligger i att underhålla sin legacy-mjukvara. Ofta utgör dessa bankers system kärnfunktioner för verksamheten och i detta fall kärnfunktioner för stora delar av branschen. Eftersom fokuset för vår uppsats å andra sidan ligger i hanteringen av teknisk skuld måste dessa verksamheters system och hantering av dessa bedömas på samma premisser.

Fortsättningsvis kan vi även se att företagen inte arbetar särskilt genomgripande med någon av teorierna från varken Li et al. (2015) eller Alves et al. (2016). Däremot kan vi se att vissa av företagen arbetar med metoder som liknar delar av teorin, framförallt gällande mätning och prioritering av skulden. Vi tror, med bakgrund av teorin, att många företag hade gynnats ifall de arbetade med de här strategierna mer utpräglat och om de på ett mer konsekvent sätt genomförde mätningar, prioriteringar och värderingar av skulderna både innan och efter skulden tas. Detta för att på ett bättre och enklare sätt kunna avgöra dels kostnad mot nytta gällande genvägar som eventuellt bör tas, samt vad det kostar att åtgärda en skuld mot vad det kostar att behålla en redan befintlig skuld.

Slutligen avser vi besvara forskningsfrågan som ställdes i det inledande kapitlet. Teknisk skuld skiljer sig åt mellan olika aktörer på bankmarknaden i egenskap av framförallt storlek och mängd. Hanteringen av denna skuld skiljer sig åt mellan olika aktörer, där framförallt företag med äldre system och hög grad legacy tycks möta större utmaningar. När vi reflekterar över uppsatsen tycker vi att vi har besvarat forskningsfrågan på ett tillfredsställande sätt och vi är nöjda med struktur, disposition och innehåll av uppsatsen. Vi anser också att vi baserat frågeställningen på trovärdigt och adekvat material samt att vi haft respondenter med hög kompetens som kunnat besvara våra frågor.

7 Appendix

7.1 Intervju 1

Transkribering av intervju genomförd 2021-04-21

J = Joakim Sahlström (CTO på Avanza)

L = Love Anderberg

O = Oscar Wallin

A = Axel Eriksson

1. L: Då börjar vi. Vad heter du?

2. J: Joakim Sahlström heter jag. Jobbar som CTO på avanza. Jag har varit här i snart 11 år och varit utvecklare i kanske 8 av dem åren. Civilingenjör i datateknik från Linköping.

3. L: Vi kan väl komma väl komma in lite på banksektorn.

4. J: Mm.

5. L: Hur skulle du beskriva att digitaliseringen påverkat konkurrensen på bankmarknaden under de senaste åren?

6. J: Pouh. För avanza är ju digitaliseringen allt. Vi kom till på grund av digitaliseringen, det är det som gör att avanza finns. Vi startade ju 1998 eller något och det var ju med idén att göra delar av bankvärlden digital. Så det har ju påverkat enormt mycket. Annars hade vi aldrig varit här liksom. Grunden i vår affärsidé är ju att vi ska inte ha en massa kontor och vi ska ju inte vara en gammal bank utan vi ska ju möta våra kunder på nätet.

7. L: Skulle du säga att psd2 och open banking har förändrat förutsättningarna för er rent konkurrensmässigt?

8. J: Jo men det har det ju. Eh, sen kanske det inte har påverkat så mycket som man trott. Asså det är ingen som vet hur PSD2 och Open banking skulle påverka, och jag är mycket teknik och inte lika mycket affär, så jag ska inte svara för utdraget men ja det har ju påverkat och för oss har det varit positivt för vi kan ju utveckla nya tjänster ovanpå det här som hjälper oss i vår affär.

9. A: Vad för typ av tjänster menar du mer specifikt?

10. J: Amen flytta hit sina värdepapper. Vi kan göra den processen enklare med hjälp av psd2. Typiskt.

11. O: Så det är positiva effekter i allmänhet då?

12. J: Ja men för vår del har det ju varit det.

13. O: Har förändringarna gjort att ni ändrat fokus eller strategier?

14. J: Asså det har ju gjort att vi börjat satsa inom området liksom men jag skulle inte säga. Vi utnyttjar ju möjligheterna som finns där. Men det har inte gjort att vi svängt som företag. Det är ju snarare att vi har hittat några nya produkter att stå på men först och främst har det ju varit hur våra användare interagerar med oss. Vi har ju fått möjligheter att hjälpa dem mer. Jag tycker vi kan ge en bättre användarupplevelse tack vare open banking och psd2.

15. L: Ska vi komma in lite på teknisk skuld då. Vi kan väl börja med att fråga om du känner till begreppet teknisk skuld?

16. J: Det gör jag absolut! Sen kan man ju prata om olika saker när man säger teknisk skuld. Vi använder ju kanske båda begreppen teknisk skuld och legacy när vi pratar på avanza. När vi pratar om teknisk skuld då pratar vi kanske om system som av olika anledningar blivit väldigt svåra att underhålla. Allt ifrån dålig kod rent ut sagt, till kanske en design som inte underhållits och då sitter man med system som kanske inte gör det som de borde göra på ett bra sätt. För mig handlar legacy mer om att man har verktyg och ramverk som är förlegade. Så det försöker vi skilja på.

17. L: Skulle du säga att ni har teknisk skuld i företaget idag?

18. J: Alla teknikbolag har legacy. Om de inte startade igår. Men så är det ju. Vi har teknisk skuld och vi har legacy, sen har vi ganska bra koll på den. Men vi kan som alla andra bli bättre på att ha hand om den. Vi har en väldigt homogen teknikstack som gör det lättare för oss att jobba med legacy då den legacyn vi har ser ganska lik ut vilket gör det lättare att jobba bort den.

19. O: Kan ni gå på någon mer specifik skuld eller legacy ni har?

20. J: Nej men vi inför ju nya arkitekturer. Vår senaste version av arkitektur tillåter deployment av applikation precis när som helst oberoende av alla andra applikationer och vissa applikationer är inte byggda på det sättet ännu. På så sätt lever våra system kvar i ett lite äldre releaseprocess till exempel på grund av tekniska lösningar som finns i botten.

21. L: Jag måste bara fråga då jag blir intresserad. När du säger det där med att legacy finns i alla system, skulle du då påstå att det finns i olika grad?

22. J: Ja, så är det ju.

23. L: Jag tänker på just företag som utvecklats mycket tidigare än er med betydligt äldre system.

24. J: Ja precis, då kan man ju sitta med legacy i form utav teknikstackar som inte ens underhålls längre. Det är inte läget på avanza, vi kör ju java rakt igenom och det vi kallar gammalt är liksom 10 år gammalt så det är ofta rätt välfungerande. Sen har vi några komponenter som är 15 år gamla, men de är inga problem att underhålla.
- 25. A: Men skulden kan uppkomma i flera olika former? Du nämde designen men också i arkitekturen att man har en förlegad arkitektur eller förlegad kod då också kanske?**
26. J: Ja men absolut. Det är på alla nivåer och teknisk skuld ser ju väldigt olika ut. Det kan ju handla om att man sitter fast med en datalagringslösning som inte håller måttet och det kan ju vara superjobbigt att byta. Ibland är det jättekomplesa monoliter som är jättesvåra att bryta upp till mindre system.
- 27. O: Känner ni att den tekniska skulden problematiseras av personalomsättning? Alltså att personal lämnar och att det kan vara svårt att överföra den kunskap de har till nya medarbetare? Eller har ni strategier för att motarbeta detta?**
28. J: Ja, vi har definitivt strategier för att undvika det. Vi har ju starka team som äger våra komponenter, så vi har ju alltid ägare till alla våra komponenter vilket minskar problematiken med teknisk skuld. Generellt sett skulle jag säga att vi inte har några system som ingen förstår, men det tar ju tid att att flytta en komponent som har teknisk skuld eller är baserad på legacy tekniker till att inte ha det och det är inte alltid värt att göra allting "super shiny" för då hinner du inte bygga allt det nya man vill få gjort så det är en avvägning man får göra. Man vill inte vara extrem åt något håll.
- 29. L: Har ni några tydliga strategier ni arbetar med för att åtgärda teknisk skuld?**
30. J: Ja asså hur man åtgärdar kommer ju bero jättemycket på den tekniska skulden men vi följer ju våra komponenter och markerar de som är gamla och legacy i princip och vi följer upp konstant hur den arkitektoniska utvecklingen ser ut. Alltså åt vilket håll rör sig Avanza, hur många av våra komponenter lever upp till den molnarkitekturen vi har. Vi har liksom en idé om hur vi vill bygga system, och då kan vi ofta se hur väl en komponent passar in i det. Så det följer vi upp ganska mycket. Och sen är det upp till varje utvecklingsteam hur de jobbar med sin legacy, i stort. Sen om vi ska göra ett stort tekniskt språng, ja de måste de fokusera på ett speciellt område.
- 31. A: När vi läst om teknisk skuld så fokusera de flesta teorierna på kostnadsanalysen av skulden, alltså att man bör prioritera skulder i en prioriteringslista, vilka skulder kostar mest och vad blir effekterna av att åtgärda dessa.**
32. J: Ja, och det är ju på ungefär det sättet vi jobbar. Vi sätter liksom inte någon prislapp på grejerna men generellt sätt gör man det klassiska diagrammet med kostnad att åtgärda på ena axeln och kostnad att ha kvar den på den andra axeln och sen ser man vilka skulder som bör åtgärdas.
- 33. L: Så en typ av prioriteringslista?**

34. J: Ja det blir ju det. Det som faller ut från det diagrammet blir en typ av prioritering.

35. L: Du nämnde också det här med att det hanteras inom utvecklingsteamerna. Era arbetsmetoder, jobbar ni med agila arbetsmetoder?

36. J: Mm, det gör vi. Får jag säga det själv har vi kommit ganska långt med det. Teamen får välja själva om det vill köra med scrum eller vad de nu känner för egentligen. Sen tycker jag vi har i organisationen i stort med det agila arbetssättet, så det handlar inte bara om att ett team ska vara agilt. Sen handlar det också om kompositionen i teamet, är affären i teamet, ja då kan man verkligen bli agil. Har man inte det, ja då tror jag det kan uppstå utmaningar.

37. L: Kan du se en relation mellan det här agila arbetssättet och att det underlättar att hantera teknisk skuld och växande komplexitet i systemet så att säga?

38. J: Ehm, ja asså jag har ju aldrig jobbat på något annat sätt än agilt. Däremot är det viktigt att teamen får ett tydligt ansvarsområde och har kraft och möjlighet att ta tag i sin tekniska skuld. Det är något som vi som organisation måste se till att teamen har. För sen kommer de utvecklande teamen veta ungefär vilka komponenter som är riktigt jobbiga att jobba med.

39. O: Förstår jag det rätt om jag säger att den här typen av decentraliserad gruppbildning, där teamen ansvarar för sina egna processer och delar, gynnar hanteringen av teknisk skuld och effektiviteten i allmänhet att ha det på det sättet?

40. J: Effektiviteten i allmänhet, absolut! Det är nyckeln om du ska bli en effektiv utvecklingsorganisation. Tycker jag iallafall. Sen vissa saker blir lättare och andra blir svårare allting är en avvägning. Men ja det blir ju lättare att få vissa saker att hända samtidigt eftersom vi är mer decentraliserade ska vi ta en teknisk sväng åt något håll så kanske den är lite svårare men ofta när man får till det bra blir det nästan lättare för då vill alla göra det liksom.

41. L: Sista frågan om teknisk skuld, är det någonting du känner att ni borde ändra på i eran hantering av teknisk skuld? Alltså är det någonstans ni ser utmaningar?

42. J: Nja, jag tror det handlar om att lära sig liksom och förstå hur dem här avvägningarna ska se ut. Ibland borde man kanske lägga ner det lilla extra för stänga ner det sista lilla. Det måste man ju göra, det tycker jag är superviktigt att man hela tiden ifrågasätter vad man gör.

43. O: Då går vi vidare och pratar lite mer allmänt om system. Vad har ni för olika system och hur är de ihopkopplade till varandra väldigt grovt beskrivet?

44. J: Om vi då pratar om det vi bygger på avanza det som är våra egna system, vi bygger ju mycket av våra system själva. Alltså sajten och allt ni ser där är ju liksom drivet av våra

egna grejer. Vi har ju ett backoffice system som har hand om skatteberäkningar, men det är inte en del av er onlineupplevelse utan där är det bara våra system som är. Där har vi en mikrotjänstarkitektur som vi byggt på sedan 2011. Vi började bygga mikrotjänster innan begreppet fanns då det myntades först 2012.

45. O: Det måste underlätta nu under open banking kan jag tänka mig? Att ni har det redan från grunden.

46. J: Ja asså det har ju varit instrumentalt för Avanza så långt tidigare. Vi har väldigt komplexa system. Vi har många teams som jobbar med ganska komplexa grejer som ofta måste integrera med varandra. Asså även när man bygger pensionssystem måste du ha, vad vet jag, kursdata för någonting för att kunna avgöra värde eller någonting. Asså det jag menar är att det blir väldigt mycket integration mellan. Då passa mikrotjänster bra. Bygger du mikrotjänster bra så har du antagligen en API-plattform för att dela funktionalitet Så får du dem bitarna på plats så hjälper det ju.

47. O: När är de äldsta systemen som fortfarande är i bruk från?

48. J: Oj, jag vet faktiskt inte. Vi har säkert system som är 15 år gamla, men vår moderna arkitektur började vi bygga 2011 och det är liksom den absoluta majoriteten av alla våra grejer. Men där ska man säga att även de system som byggdes 2011, hur ska man säga, de har inte stått stilla sen dess utan man bygger ju vidare på mikrotjänster. En stor fördel med att ha mikrotjänstarkitektur är att det är ganska lätt att ändra i so, kan ha system som lever framåt och det är ju superviktigt.

49. O: Ja men precis, det har vi förstått också.

50. J: Andra tjänster börjar ju bli mindre viktigt på något sätt eftersom det som binder ihop alla tjänster som kommunikationslager, APIer, alla dem där blir ju superviktigt, hur delar vi datafunktionalitet.

51. O: Och du nämnde det lite kort tidigare att ni skriver i Java men använder ni av några andra programmeringsspråk?

52. J: Nej men vi kör ju.. Oj, jag kan ju inte frontend stacken så bra..

53. L: Nej men vi tänkte framför allt på backend.

54. J: Ja men i backend är vi en ren javashop. Vi tittar väl på python när det gäller ML, alltså hög nivå datahantering så att säga.

55. L: Vi var inne lite på det innan med relationen, men kan du se någon relation mellan java som programmeringsspråk och hantering av teknisk skuld?

56. J: Ja asså java är ju inte jättemodern om man ska vara lite hård mot sig själv. Men skulle Java fortsätta sin utveckling som de gjorde fram till version 8, då det gick fruktansvärt

långsamt under många, då hade jag varit mer orolig än vad jag är nu när de faktiskt fått fart på språket igen.

57. L: Men kan du se en koppling mellan teknisk skuld och java som programmeringsspråk, alltså att det underlättar för er hantering av teknisk skuld?

58. J: Nej men det har ju underlättat hanteringen eftersom allting ser likadant ut. Ska vi lyfta något, ja men då är det likadant överallt och det är ju mycket lättare än att lyfta 14 olika grejor.

59. A: O Java är väl också ett så pass flexibel språk kanske, så ni hade inte haft jättestora fördelar av att använda er av exempelvis python kanske?

60. J: Nej inte i backend stacken iallafall. Sen så här, java är kul som programmeringsspråk och det är bra, asså det funkar ju bra. Sen det man inte ska glömma är ju floran av alla tredjepartsverktyg runt om, alltså alla bibliotek o så.

61. L: Tror du att det finns någon slags förändring i vad man värdesatte då och nu vid utveckling av system?

62. J: Ja asså kollar man på Avanza så värdesätter vi förändring mer nu och att kunna förändra grejerna. Det är så otroligt viktigt. Asså det vi inte visste inom Avanza när vi höll på att bygga det 2007 var ju hur vi bygger ett system som vi om 3 år kan förändra. Det är sånt jag tycker vi har lärt oss. Vi har också bytt teknikstack så nu har vi helt andra sätt att lagra och hantera data än vi hade tidigare.

63. A: Hur menar du att man bygger ett system som är lätt att hantera 3 år senare?

64. J: Ja men asså mikrotjänstarkitekturen är ju svar på det. För då minskar ju scopet i dina applikationer och mindre applikationer är ju lättare att förändra, dem är ju lättare att bygga om. En annan viktig sak med att bygga mikrotjänstarkitektur är att du ska ju gömma datan, du ska gömma den interna datarepresentationen mellan mikrotjänsterna så mikrotjänst A ska ju inte veta något om hur mikrotjänst B hanterar datan. Och det gör ju att du kan ändra mikrotjänst B, för det finns inget beroende när du delar din databas exempelvis som du gör med gamla system då är det jättesvårt att ändra datan. Och nu så har vi istället, det som är gemensamt är ju då dem APIerna som tjänsterna kommunicerar via. Och då kan man ju titta på andra frågeställningar där, då kan man ju titta på hur vi ser till att dem här kan evolutioneras framåt.

65. O: Det är väl en av de stora fördelarna med mikrotjänstarkitektur att det är såpass flexibelt gentemot andra arkitekturer?

66. J: Absolut, flexibilitet och ja det blir ju i flera dimensioner. En dimension är ju att ändra saker och en annan är just skalning. Vissa av våra system måste vi skala upp i 130 instanser och då kan vi ju göra det medans andra inte behöver vara så uppskalade. Den

problematiken är ju mycket enklare än att försöka sig på en jättemonolit o se till att den fungerar bra.

67. O: Ser ni några nackdelar med användandet av mikrotjänster?

68. J: Jo men det är klart det gör det, man stöter på rätt mycket det är ingenting man börjar med och kör. Vi har ju varit med på resan länge så vi har ju löst många av våra problem. Det första man stöter på var manageringen av de här grejerna. Avanza, vi har 1400 mikrotjänster och 6000 instanser, det är liksom ingenting man bara gör. Det häre ska ju funka jämnt, så hur lyckas man deploya en sån miljö varje vecka eller till och med oftare, bara den frågan är ju svår. Hur gör man med övervakning av de här systemen? Hur gör man med feltolerans? Det är ju en superviktig fråga. Nu är ju det löst, det var ju nån, om det var någon på netflix som började lösa de där. Men man måste ju ha, vad gör man om en mikrotjänst går ner. Har du så många program som vi har då kommer det gå sönder och det går sönder saker jämt och det måste du hantera och det ger ju en lite mer komplex programmeringsmodell. Så det är ju sånt man stöter på, och vad blir det mer.. det är klart det blir ju andra type av utmaningar vid deployment och så vidare. Det gäller ju att bygga tjänster så att de inte har för hårda kopplingar till varandra.

69. A: Hårda kopplingar, hur menar du då?

70. J: Ja men säg att tjänst A kräver att tjänst B är igång för att den ska fungera. Bygger du fel så kanske du bygger så att tjänst A den startar inom tjänst B inte är igång. Tillåter du att det är så i din arkitektur kommer du förr eller senare ha ett system som inte går att starta. Så det är mycket sånt man måste tänka på. Och sen hur asså, det var ju en lärdom avanza dragit de senaste 10 åren vi kommunicerade via serialiseringen av meddelanden i våra APIer var ju baserade på Javas serialisering och det gör ju att då blir klient och server ganska hårt bundna till varandra så dem måste ju ha samma version av javaklassen, då de måste vara kompilerade med samma kodversion. Och då kan du ju inte stegvis uppdatera systemet utan då måste du uppdaterar delar av systemet parallellt. Det blir ju en jättestor svårighet, speciellt när organisationen börjar växa när det helt plötsligt är 50-70 utvecklare som sitter och bygger systemen. Men där hjälper ju mikrotjänster till, att gör man det rätt, som vi tycker att vi gör nu, så får man inte dem problemen.

71. L: Nu börjar vi ju närma oss dem här 30 minuterna, men jag kom bara tänka på något annat angående teknisk skuld. Vi har ju förstått att det kan uppstå en teknisk skuld när man tar genvägar i programmeringen för att hinna deadlines med mera för att snabbt anpassa sig till förändringar och så vidare. Kan du se en relation mellan att andra avdelningar pressar på dina team att skynda på utvecklingsprocessen då de inte har samma förståelse för teknisk skuld?

72. J: Ja men precis. Nej men vi jobbar ju inte så. Vi har ju fått in affären i teamen. Eller det är klart, det kan finnas såna utmaningar, ibland är det ju regulatoriska grejer. Alltså vi är en bank vi måste göra såhär, ibland är det en börs som ändrar deras kommunikationsprotokoll eller någonting. Det är klart att den problematiken finns där vi

försöker ju undvika den så mycket som möjligt. Kvalité är ju väldigt viktigt. Man får ju alltid betala om man inte levererar med bra kvalite

73. O: Vi kan väl avsluta med att om man kan få någon slags spekulativt svar gällande bankmarknaden i allmänhet nu efter psd2 och liknande, hur kommer de större aktörerna, alltså storbankerna påverkas utav det?

74. J: Jag är ju inte affärsmänniska men min generella spaning är ju att små uppstickare gör en specifik sak väldigt bra och det verkar ju storbankerna ha svårt att möta upp på.

75. O: Ja men tack för det. Man får bara bocka och buga, det här har varit otroligt intressant och jag tycker vi fick ut mycket mer än vad vi hade kunnat förvänta oss. Stort tack verkligen.

76. J: Ja, jag ser fram emot att se hur det går för er.

77. O: Du får ha en riktigt trevlig dag, och tack så mycket ännu än gång.

78. L: Tack så mycket!

79. A: Ja tack!

80. J: Ha det bra!

7.2 Intervju 2

Transkribering av intervju genomförd 2021-04-30

O = Oscar Alvarsson (CTO på Rocker)

L = Love Anderberg

OW = Oscar Wallin

A = Axel Eriksson

1. **OW: Vi kan börja lite mer brett då, hur skulle du beskriva att digitaliseringen påverkat konkurrensen på bankmarknaden de senaste åren?**
2. **O:** Nä men det är väl ganska självklart att den ökat, konkurrensen. Det går ju att göra väldigt många saker inom banksektorn som inte gick tidigare. Med tanke på att du tidigare var knuten till rent fysiska upplevelser på bankkontor och liknande. Med digitalisering får du ju helt andra möjligheter att kunna kontakta... Kunna ha rådgivning, interaktion med kunder som inte är knutna till fysiska platser liksom. Det är väl det korta svaret.
3. **A: Skulle du säga att PSD2 och Open banking har förändrat era förutsättningar rent konkurrensmässigt?**
4. Ja absolut. Det har det ju, vi är ingen bank utan ett betalningsinstitut men det här är en stor förutsättning för att vi ska kunna erbjuda våra tjänster till konsumenter. Det har ju funnits lite innan till exempel Tink var ju väldigt tidigt ute redan 2013 med den här typen av reverse engineering grejor. Eller de här screenscraping-metoderna som en del har kört. Så att det har ju funnits där tidigare men det blir ju lite mer rumsrent genom PSD2 och Open banking. Det är ju verktyg som är framtagna för att förbättra för konsumenten. Och det är ju det vi vill göra så det är grymt att man kan aggregera in bankkonton och kort och så på en plats, till exempel Rocker då. Där du sen kan lägga, en mängd tjänster ovanpå det, som du inte har i din vanliga bank. Jag har länsförsäkringar till exempel. De får jag in i rocker appen. Jag kan göra saker med transaktioner och kontot i Rocker appen som jag inte kan göra på LF, och det tycker jag är ganska häftigt. På sikt är det ju sådana saker som driver kanske en förflyttning från min gamla bank. Så för vår del så tycker jag det bara är bra. Jag kan se redan där på fråga 2 om det funnits några dåliga effekter och jag kan inte komma på riktigt, för vår del tror jag det är uteslutande bra.
5. **A: Ni har inte heller en direkt konkurrenssituation mot bankerna?**
6. **O:** Nä det har vi ju inte på det sättet. Inte med lönekonton. Swish och mobilt bankID är det ju storbankerna som utfärdar och det är ju bland annat sådana saker som jag tror håller folk till storbankerna. Men när det luckras upp blir vi ju konkurrenter mot de på många fler plan, fler plan än idag. Den största konkurrenssituationen för oss i närtid är ju liknande aktörer som oss, alltså andra fintech aktörer medans storbankerna ligger längre bort.
7. **L: Om vi går vidare lite och nischar ned oss till lite mer specifikt det som vi tänkt undersöka nämligen teknisk skuld. Känner du till begreppet?**

8. O: Tyvärr ja.

9. L: Skulle du säga att er verksamhet är påverkad av teknisk skuld?

10. O: Ja, alltså jag vill hävda att alla verksamheter som skulle säga något annat skulle ljuga. Det är min spontana reaktion på det där. Och inte minst i den världen som vi är i. Det är alltid en balansgång när det kommer till snabbväxande aktörer som ska växa fort, ut med nya produkter snabbt. Så det är klart att det kommer finnas kompromisser, med saker som måste ut på marknaden här och nu, och problem som man får återkomma till och förbättra lite senare. Det är en naturlig avvägning för oss liksom. Så skulden finns där och den hanteras absolut men går inte att undvika till fullo.

11. OW: Mm, och man kan utnyttja det då också låter det som att man kan få ut produkter snabbare genom att ta dessa genvägar och avväga vilka man kan ta.

12. O: Ja alltså det finns en del genvägar man inte kan ta i den värld vi agerar i. Det är en förtroendebransch och säkerhetsaspekterna går liksom inte att tulla på. Men det kan ju finnas andra saker som man kan kompromissa med, för att vara snabbare ut liksom. Det kanske inte är så modulärt som man önskar liksom det kanske inte är så enkelt att flytta över det i en annan kontext. Vi kanske är lite mer beroende mot den tredje parten än vad vi önskar att vi var. Så att absolut, avvägningar finns ju där.

13. L: Vad skulle du säga är en stor bidragande del till er tekniska skuld? Ett vanligt förekommande moment som ger teknisk skuld?

14. O: Ja men det är väl klart att det finns områden som kanske varit med från start, som har varit med längre. De skulle kunna ha ett större element av det där. Och där man i ett tidigare skede kanske hade andra ingångsvärden om vad som värdesätts. Man kanske fokuserar på en typ av produkt eller en typ av marknad och sen när man har fler produkter och går ut mot fler marknader då kan det ju finnas skuld relaterat till det.

15. A: För att dessa funktionerna är anpassade till en annan målgrupp eller?

16. O: Ja, eller en annan sammanhang just då eller under andra omständigheter. Finns ju andra större faktorer kanske ändring av målbild och så det kan ju gå lite fortare när man är ett mindre och yngre bolag och strategin formas ju utefter det. De är ju inte fasta på 5-års planer alla gånger utan kan ju ändras.

17. OW: Kan du ge ett exempel på hur ni arbetar med teknisk skuld, om ni har några särskilda strategier, rutiner eller best practices?

18. O: Ja alltså dels är det ju det här. Vi jobbar ju agilt, vi jobbar i sprinter med 2 veckors cykler med planeringar och standups och allt det här som ni säkert är bekanta med, och där kommer ju det här in också som en del av det. Det finns ju alltid en grej med att få in maintenance som en del av sprintarna också och vi har även haft rena sprintar som inte

fokuserat på nya features utan bara ta hand om eventuell skuld som har funnits. Så det är väl den approach vi har. Vi delar i backloggen som alla andra saker som sedan hanteras av våra team och utvecklare.

19. L: Så med andra ord ni arbetar med någon slags prioritering av den här tekniska skulden. Men kanske inte som en separat del där ni bara har teknisk skuld utan det hamnar tillsammans med den andra på backloggen?

20. O: Ja exakt, precis.

21. L: Skulle du säga att det finns något specifikt sätt ni mäter eller utgår ifrån när ni kollar på teknisk skuld? Kollar ni till exempel på kostnad mot effekt av att återbetala skulden?

22. O: Ja de tankarna finns ju alltid där. Cost-benefit-delen är ju central i allt så att säga. Särskilt ifall man har kanske begränsad kapacitet i ett mindre bolag och olika typer av fokus på vad som är viktigast för stunden.

23. OW: Mm...

24. OW: Nu är ju ni ganska ny aktör, och komplexitet i systemen kanske ni har men ni har kanske inte så mycket legacy i mjukvaran ännu. Men finns det en koppling mellan komplexiteten och teknisk skuld, att ju mer komplext ju svårare är det att hantera den tekniska skulden.

25. O: Ja men visst är det så, de mer komplexa sakerna har väl en större risk att dra på sig mer skuld också, och även där driver ju tidsfaktorn på det. Vi har ju några grejer t.ex konsumentlån som vi lanserade 2017 så vi har ju funnits på marknaden sedan dess.

26. OW: Det är som Bynk då eller?

27. O: Ja precis det är under varumärket Bynk. Och det är klart, det är ju en ganska komplex värld och den har varit med från start då vi arbetade i mindre team. Och komplexiteten där kan ju såklart på sitt vis driva saker. Men å andra sidan har vi produkter som vi byggt efter det som också har en hög komplexitet och tiden kanske inte varit så lång. Men om man bara fortsätter på samma sätt som man gjort innan så kommer ju samma skuld tillbaka även i de nya produkterna senare så det vill ju till att man lär sig av det man har gjort tidigare så man reducerar mängden skuld till nästa gång.

28. L: Ni är ju ett väldigt digitalt bolag, som du själv uttrycker ett "fintech-bolag" och skiljer er på många sätt från en traditionell aktör på bankmarknaden. Men kan ni se en tidspress från andra avdelningar som leder till en större teknisk skuld då de inte har samma insikt i det tekniska arbetet? Eller är det en fördel för er att ni är så digitala att det lite är inpräntat i kulturen?

29. O: Ja jag tror vi har stora fördelar av att vi har börjat där, som ett digitalt bolag och vi har ju allt i google cloud platform vilket gör ju att vi har bra förutsättningar genom det såklart. Dessutom moderna tekniker och inte så lång historik. Samtidigt har vi ju många stakeholders som driver på och vi har ju självklart deadlines som vi måste leva upp till och den tidspressen kan ju absolut driva på teknisk skuld.
- 30. OW: Yes, ska vi övergå till system och arbetsmetoder eller har ni någonting mer att tillägga?**
- 31. L: Nä men man kan väl bara avsluta med att fråga ifall du själv känner att det finns någonting i hanteringen av teknisk skuld som ni borde ändra på?**
32. O: Nä men jag tror väl att det är fortsatt viktigt att man har ett kontinuerligt lärande i det där. Hur vi hanterar det så när vi gör nya saker att vi alltid hanterar det lite bättre varje gång och gör den tekniska skulden lite mindre för varje gång. Det är väl mest det och att vi har en uppföljning och medvetenhet i våra tech-teams. Sen är det ju också viktigt från ett management håll att man har förståelse för det där och att man tillåter sig ta beslut och kompromisser och då måste ju också hanteringen komma med i prioriteringen längre fram så att det faktiskt hanteras.
- 33. A: Om vi går vidare då till lite mer om system och arbetsmetoder. Vill du bara beskriva väldigt grovt vad ni har för system idag och hur de är kopplade till varandra?**
34. O: Ja, google cloud platform är ju centralt för arbetsmetoden. Sen har vi ju en IOS och en Android app för våra användare. Under det så ligger det ju en front- och backend som hanterar all kommunikation mot underliggande microservices, så vi har ju en mikrotjänstarkitektur.
- 35. A: Ok, och ni utvecklar mikrotjänsterna själva?**
36. O: Ja det gör vi ju. Men vi använder såklart tredjeparter för våra tjänster där det finns bra sådana. T.ex då bankID, Swish, Trustly och Tink eller vad det nu kan vara, alltså de delarna som jackar in i Rocker-upplevelsen.
- 37. L: Om vi pratar om de här backend-systemen. Ungefär när är det äldsta systemet ifrån, som fortfarande är i bruk?**
38. O: Ja, vi lanserade ju i 2017 och började dra igång i 2016 så det är ju inget av det som vi byggt själva som är äldre än så. Sen så klart använder vi ju partners som har äldre core-system än så, t.ex på lån/kredit-sidan, spar-sidan eller vad det nu kan tänkas vara där det finns andra aktörer med andra typer av core-banking system som ju kan ha annan livslängd.
- 39. L: Vilket språk är era system skrivna i?**

40. O: Ja, alltså scala är det primära språket för våra backend-team sen är det ju react/JS för frontend-teamen.
- 41. L: Kan du se någon koppling mellan er hantering av teknisk skuld och det språket, och ett så pass ungt system? Positiva eller negativa konsekvenser.**
42. O: Ja alltså det är klart att det är positivt att det är ett ungt system på det sättet och verktygen för det och förutsättningarna att arbeta med det blir lättare med en moderna arkitektur. Så för vår del är det en väldigt positiv sak.
- 43. OW: Kan ni se att digitaliseringen har påverkat vilka faktorer som ni värdesätter i era system? Trots att ni är ett relativt ungt företag. T.ex om det är effektiviteten, användbarhet eller kanske flexibiliteten med mikrotjänster då inte minst.**
44. O: Hmm, alltså.. ja för det första finns vi ju till tack vare den utvecklingen som varit i och med digitaliseringen. Med den affärsidé och det upplägg vi har är digitaliseringen en förutsättning för oss. Sen så försöker vi såklart ligga i framkant med att ta till nya grepp allt eftersom det som kommer genom digitaliseringen. Vi försöker ju driva på det själva såklart men också använda oss av den innovation som kommer som en del för att förbättra vår Rocker-upplevelse. Så digitaliseringen följer vi ju tätt för att kunna ta nytta av.
- 45. L: Med andra ord låter det ju som att kunna anpassa systemen blir av allt större vikt allt utifrån nya tekniker eller digitaliseringsmöjligheter.**
46. O: Ja absolut, och alla de orden du nämnde flexibilitet, effektivitet och användbarhet är ju av stor vikt för olika delar i systemen och där finns det ju nya möjligheter i och med digitaliseringen.
- 47. OW: Ja, vi kan gå vidare då med en fråga kring mikrotjänstarkitekturen som du tidigare nämnde. Varför har ni valt att ha en sådan arkitektur?**
48. O: Vi har ju valt det för att vi helt enkelt tycker den arkitekturen passar oss eftersom den tillåter oss att vara mycket mer snabbvärliga. Att kunna släppa nya produkter eller features fortare. Och möjligheten att kunna ha mindre teams som arbetar fokuserat på sina services. Möjlighet att arbeta med olika programspråk i olika services och tredjepartsintegrationer som en del av det. Skulle vi behöva byta ut saker behöver det inte vara ett jättestor projekt utan vi kan byta ut tjänster löpande. Det är väl några av de fördelar vi ser med mikrotjänster.
- 49. OW: Ja men precis, det är väl det vi också själva har förstått att mikrotjänstarkitekturen är mycket mer flexibel.**
- 50. L: Kan du se en koppling mellan mikrotjänstarkitekturen och hanteringen av teknisk skuld? Att den eventuellt är lättare att hantera i och med arkitekturen?**

51. O: Ja jag skulle säga att det är både och lite beroende på hur man ser det men jag skulle ändå säga övervägande att det är bättre för vår del. Sen kan det såklart bli komplext när det blir väldigt många services med beroenden dem emellan och så att ha en komplett översikt av alla delar och var skulden finns. Det går att argumentera för det i alla fall.
- 52. OW: Ja och mikrotjänster har väl i och med Open banking blivit ännu mer premierat med tanke på att det krävs en del för att man ska hinna med i utvecklingen?**
53. O: Ja lite som vi var inne på innan, det blir ju lite lättare att hantera och man blir ju lite mer snabbriktig i och med att man kan ha flera team som jobbar på olika services. Vi blir ju mer snabbriktiga både på tekniksidan och på funktionalitetssidan och som sagt plocka in tredjepartsfunktioner eller byta ut de mot någon annan funktion.
- 54. OW: Så med andra ord hade ni använt er av mikrotjänster även ifall Open banking inte varit i laga kraft så att säga?**
55. O: Ja absolut det hade vi.
- 56. OW: Ja.. vi har egentligen inte så mycket mer vi vill avsluta med en högst spekulativ fråga och det finns ju inget rätt svar riktigt men du som ändå arbetar i den här branschen hur ser du själv på den framtida banksektorn? Hur tror du den kommer präglas av digitalisering och fintech mot de mer traditionella aktörerna?**
57. O: Ja men det blir spännande att se vad som händer. Storbankerna och bankerna kommer ju att vara kvar, såklart över en överskådlig framtid. Det är inte så att någon aktör kommer försvinna över en natt på något sätt. Sedan kan man ju ställa sig frågan vilken roll de olika aktörerna eventuellt kommer ha, vad man ska vara bra på. Det går ju att prata om de olika lagren att det finns ett lager av aktörer som är väldigt duktiga på infrastrukturen kring allt inom bankvärlden och sen finns det ett annat lager av aktörer som är väldigt duktiga på interaktionen med användaren och har sin nisch där. Och så finns det ju samarbeten där emellan de olika aktörerna som är ju liksom avgörande för respektive part om man lyckas eller inte.
- 58. OW: Så ni behöver varandra kan man säga.**
59. O: Ja men absolut, jag tror att det finns plats för många aktörer inom det här området i framtiden. Vi får väl se. Det finns ju även big-tech med de stora tech-bolagen som har ett saying i allt det här. Som också har gett sig in i det här området och har såklart enorma muskler till att göra det. Vi har ju sett hur konkurrens hanteras och vilka typer av regulativa faktorer och så som kommer in, som en del av det, vad man får göra och inte göra i det space som också kommer att driva utvecklingen. Så det finns såklart många delar i det som är väldigt spännande. Men jag tror väl inte i alla fall på närtid att det kommer bli färre aktörer utan kanske snarare fler, ett tag till. Det vi inte har sett, som vi har sett i andra branscher att det är mycket uppköp och att branscher konsolideras. Det har

inte skett inom fintech och banking världen i någon större omfattning ännu. Det kanske kommer, vi får se.

60. OW: Oscar, det har varit fantastiskt lärorikt att få intervjua dig. Snällt av dig att du ville ställa upp och låta oss göra det. Det uppskattar vi verkligen.

61. O: Kul! Hoppas det var något av värde för er som ni kan använda er av.

62. OW: Definitivt! Mer än vad vi kunde förvänta oss. Tusen tack! Vi hör av oss också när vi är klara så du kan få se rapporten också. Hoppas att du kommer tycka det är intressant.

63. O: Absolut, det blir kul. Ha det så bra!

64. OW: Tack så mycket, ha det gott!

7.3 Intervju 3

Transkribering av intervju genomförd 2021-05-02

X = (Software developer hos en av storbankerna)

L = Love Anderberg

OW = Oscar Wallin

A = Axel Eriksson

1. L: Vad heter du och kan du berätta kort om vad du arbetar med?

2. X: Jag heter X och jag jobbar som Cobol- och Javautvecklare här på "storbank". Så jag jobbar med att utveckla olika betalningssystem, förvalta de system vi har och helt enkelt göra en utveckling utefter de affärskrav som kommer.

3. L: Kan du berätta lite om din bakgrund?

4. X: Jag har pluggat systemvetenskap och innan dess har jag jobbat lite med marknadsföring. Programmering har jag hållit på med i ungefär 5-6 år och jag har arbetat ca 1 år på *storbank*.

5. L: Vi kan börja lite övergripande över banksektorn, hur skulle du beskriva att digitaliseringen har påverkat konkurrensen på bankmarknaden de senaste åren?

6. X: Extremt mycket, vilket hör ihop med ganska många saker. En grej är att det blivit mycket lättare att få bolån. Det har blivit betydligt enklare att ansöka online eftersom det finns mycket mer information för kunderna. Och särskilt i samband med coronapandemin så tror jag inte det är lika viktigt längre med en personlig bankman som man känner, vilket det var för. Samtidigt som pandemin har gjort att man kanske är mer benägen att lita på folk över internet och ta de här större besluten över internet, så har det även hänt att många har låga bolåneräntor nu i samband med att det är så låg reporänta, vilket gör att många nya aktörer kan locka med extremt låga räntor. Hela den mixen av att det är lättare att söka upp information, det är lättare att söka bolån, det är lättare att starta ISK-konton, bidrar till att blir det en helt ny spelplan. Och i samband med Open banking och PSD2 blir det helt enkelt en stor förändring. Samtidigt är en trygghet av att jobba på en storbank att man ändå har så pass hög kreditvärdighet vilket gör att det blir betydligt billigare att låna pengar av riksbanken, man får lägre ränta osv än nyare uppstickare på marknaden som kanske nischer in sig på andra saker. Så även om den nya spelplanen gör det lättare att gå till andra aktörer, så blir det också lättare för de större etablerade bankerna, eftersom de kan konkurrera på det sättet. Många vill ju fortfarande låna till bostäder där vi större aktörer har en del fördelar. Sen kan man även säga att tidigare var det lite mer "one-stop-shop", att man hade ett bolån, sitt sparande osv på en bank. Men nu känns det mer som att många har flera olika aktörer, tex att man har sitt sparande via Avanza och Nordnet då båda har nischat in sig på just dessa segment och därmed har bättre gränssnitt och snyggare lösningar kring detta.

7. A: Menar du att det numera är vanligare att använda sig av olika aktörers för olika typer av tjänster?

8. X: Ja precis, det kan nog bero mycket på att vi blivit mer medvetna om att digitaliseringen faktiskt gjort att det blivit så här.
- 9. L: Hur skulle du säga att PSD2 har förändrat förutsättningar för er? Har det haft positiva eller negativa effekter?**
10. X: Det är så klart blandat. Dels finns det positiva effekter eftersom vi kan integrera annat, alltså andra bankers tjänster osv. Man kan erbjuda sina kunder att samla allt mer hos oss på vår plattform, tex ge saldo från andra banker, hämta transaktioner från andra banker osv och det är självklart positivt. Men sen går det åt andra hållet också. Eftersom det numera är lagstadgat så måste man ju utveckla öppna APIer för att folk ska kunna komma in och ta del av info osv, vilket då gör att andra aktörer kan ta del av vår infrastruktur. Men samtidigt hände detta redan på olika sätt förut, bara inte i samma utsträckning. Tink exempelvis, de bytte i och för sig affärsinriktning för något år sen, men de använde sig av, ja inte öppna APIer, men en slags teknologi, jag tror det var screen scraping, där de liksom kollade runt och hade ganska mycket av den informationen som finns just nu också. Och det var ju så att Tink var en väldigt välanvänd tjänst som många använde, även om de just då inte direkt lockade bort kunder från banken utan istället var som ett slags komplement. Alltså har sådan här typ av teknik funnits längre än bara PSD2 och Open banking. Men självklart har våra förutsättningar förändrats rent konkurrensmässigt på grund av regelverksförändringarna. Nya aktörer kan komma in på marknaden och egentligen bara fokusera på de mest lukrativa verksamhetsdelarna och dra mycket nytta av den infrastruktur som banker redan byggt upp.
- 11. O: Har dessa förändringar gjort att ni ändrat fokus/strategier eller arbetar ni på liknande sätt som tidigare?**
12. X: Det här är inte riktigt mitt område då jag arbetar mer tekniskt, men jag tror att det drivs mycket av lagkraven. Det är så hårda krav från FI att man måste till tillgodogöra det här och samtidigt har vi väl insett att man kanske inte behöver fokusera på att vara precis bäst på allting, utan att det är naturligt att små och nya företag som är anpassade efter PSD2 och Open banking är mer snabbrikliga osv.
- 13. L: Känner du till begreppet Teknisk skuld?**
14. X: Absolut, det är en central del av vårt arbete så det känner jag till väl.
- 15. L: Ja det kunde vi tänka oss och där kommer vi in på vår nästa fråga, är er verksamhet påverkad av teknisk skuld?**
16. X: Ja dels är det ju äldre system, så det kan tex vara mycket småfixar som kanske något temporärt krav som blivit permanent. Det är mycket gamla lösningar och gammal arkitektur. Det är ju system som liksom byggts på varandra, man kanske börjar med en fix här och där och sen kanske det kommer något slags lagkrav som gör att vi måste

komplettera med det här osv. Så istället för att skriva om allting så utvecklar man en del av det och menar att det är bra nog. Sen blir det också en prioriteringsfråga, om man har något som ska åtgärdas, men så sitter man med andra deadlines också, och så finns det dessutom ett lagkrav som säger att det här måste fixas innan 2022, då blir det enklare för utvecklare att ta en snabbare lösning och göra på ett visst sätt då man vet att det går snabbt och att det fungerar, istället för att gå in på djupet och undersöka den absolut bästa lösningen: Man kanske heller inte får den budgeten eller tiden att göra de bästa lösningarna för saker som redan fungerar. Sen är det väldigt jobbigt i allmänhet att arbeta mot teknisk skuld. Dels är det väldigt tråkigt, det kan ju va som så att man vet hur saker fungerar och man vet att det här borde vi göra om eller det här borde vi göra bättre, men det känns inte så bra att arbeta med saker som redan existerar. Det är inget kunder ser och det är inget som uppmärksammas internt kanske, åtminstone jämfört med att utveckla en helt ny feature kanske som får massa cred och som därmed blir roligare bekvämare och inte så jobbigt.

17. L: Kan du se en koppling mellan att ni sitter på äldre system och äldre arkitektur och att det påverkar den tekniska skulden?

18. X: Absolut. Dels är det ju att det finns så extremt många samband mellan de olika systemen, och dels är det ju att det är äldre system och det är ingen hemlighet att man inte har samma möjligheter att jobba med DevOps och jobba med continuous integration. Enhetstester kanske inte heller finns på samma sätt och det tar längre tid att utveckla saker och skriva om saker, och man kan inte automatisera testandet på samma sätt.

19. A: Använder ni er av särskilda strategier för att hantera teknisk skuld?

20. X: Jadå, en strategi är tex att så fort man ska in och ändra i något gammalt så ska man också gå igenom det och se till så att det inte finns någon gammal eller förlegad kod. Man ska alltså fixa till allt som rimligt kan förbättras. Men det är oftast inget aktivt arbete kring det som till exempel att vi skulle ha särskilda sprints ägnade åt att enbart hantera teknisk skuld. Istället blir det mer inbakat i att om vi ändå ska in och ändra i det här programmet så ser vi till att försöka förbättra det också.

21. L: Kan du gå in lite mer specifikt på hur skulden uppkommer och om det finns några specifika skulder hos er?

22. X: Hmm, ja låt mig tänka lite bara... Dels hade det kunnat var att man i äldre system har skickat ut vissa felmeddelanden där man larmat om vissa saker, och i slutändan är det en väldigt manuell handläggning att hela tiden kolla om det här verkligen är seriöst eller inte, istället för att gå in från början och skriva om det här programmet till att det kanske inte måste larma om allting. På så sätt har det blivit en mycket större tid man drar ut det på åren att behöva kolla på de här sakerna manuellt och liknande. Så det är väl ett tydligt exempel att om man hade gjort det rätt från början så hade det varit enklare att hantera i längden. Samtidigt måste man ta i beaktning att man inte alltid kan veta det initialt när man utvecklar en funktion. Men det leder ju till att tiden går och att det kanske inte blir prioriterat att skriva om det här i längden.

23. A: Så det är mycket sådana här mindre saker som förekommer ofta?

24. X: Ja precis, det är sällan det är särskilt allvarliga saker som blir vår tekniska skuld egentligen utan det är mest att det är jobbiga och tidskrävande saker.

25. O: Du nämnde tidigare att det kan uppkomma tidspress från utomstående faktorer som t.ex. lagförändringar och liknande, känner ni även av tidspress från andra avdelningar som kanske är mindre tekniskt lagda och därmed inte har samma förståelse kring vikten av vissa tekniska problem eller hur tidskrävande de är osv?

26. X: Ja absolut, så är det verkligen. Det finns en hel verksamhets sida som egentligen har varit det viktigaste i banken och som vi bygger systemen för. Det kan ju bli svårt när de frågar varför något inte fungerar eller varför något fungerar på ett visst sätt osv. Särskilt eftersom att oavsett om det är en mindre fix att kolla på så kommer det krävas testning, det ska kodgranskas, det ska genom flera processer där folk ska kolla på det och allt ska dessutom planeras in och lyftas osv. Med andra ord saker som de kanske inte alltid har i åtanke. Det finns alltså så stora tekniska krav på att allting ska bli rätt, även för väldigt små ändringar. Så tidspress finns ju alltid där eftersom det finns så himla många olika intressenter som har krav och som alltid ska prioriteras in.

27. L: Ni är ju en rätt traditionell organisation som har funnits länge, och i grunden är ni inte en digital organisation så att säga. Så appropå det Oscar var inne på, skulle du säga att detta ökar er tekniska skuld på något sätt, då stora delar av organisationen kanske inte har just den tekniska förståelsen?

28. X: Ja men så är det verkligen. Affären är ju från början driven från bankkontor och det är de behoven som funnits där från början, man började ju liksom med papper och penna och en checkbok och sen utvecklades IT-systemen för att stödja organisationen. Med andra ord är systemen anpassades efter hur verksamheten såg ut då och utvecklingen drevs på efter vad den tidens intressenter behövde, alltså människor som arbetade på banker, kontorschefer, handläggare, rådgivare osv. Så absolut är det en nackdel i jämförelse med nyare aktörer.

29. L: Så man kanske kan säga att systemen utvecklades som stöd för verksamheten tidigare medan de idag utvecklas med mer fokus på användbarhet för kunden?

30. X: Ja precis. Det byggdes inte från slutkundens önskemål från början utan istället var det kanske rådgivare på bankkontoren som ville kunna göra affärer digitalt för kundens räkning och sen har det byggts på och fortsatt. Det hör ju också ihop med den tekniska skulden att de här systemen är så gamla och att de har den här funktionaliteten, eftersom de kanske uppkom på 90-talet då kundens affär inte såg ut på alls samma sätt som den gör idag. På den tiden kunde man kanske inte ens logga in själv osv.

31. O: Du som bl.a. är Cobolprogrammerare, finns det någon koppling mellan programmeringsspråk och teknisk skuld?

32. X: Så är det verkligen. Språken har ju utvecklats, det har i och för sig Cobol också, men det är ingen hemlighet att om man hade utvecklat ett nytt system idag så hade det med stor sannolikhet inte varit just Cobol. Det har och göra med att man inte riktigt kan göra enhetstester på samma sätt, det är krångligare att ändra saker, det är väldigt mycket mer manuell handpåläggning när man ska göra om saker osv. Så absolut finns det en koppling mellan de.

33. A: Hur gäller det med bibliotek och så, kan man hitta mycket online om Cobol?

34. X: Nej om man jämför med andra språk så finns det väldigt bristfälligt med dokumentation men istället är det mycket kunskap som finns internt inom organisationen. Det är ju oftast hos bankerna och försäkringsbolagen som man finner Cobol och därför finns det inte riktigt någon "Open source" känsla kring det, utan alla har försökt hålla informationen för sig själv. Och sen tror jag det finns väldigt få människor som sitter hemma på fritiden och utvecklar Cobol-system så på nätet är det svårt att hitta. I så fall är det IBMs officiella hemsida som gäller eller vissa "super-users" så att säga som man finner på nätet, men annars är det mycket att man finner informationen och dokumentation internt och att man får utgå mycket från vad kollegorna säger.

35. L: Hur ser det ut gällande dokumentation till de äldre systemen? För som vi har förstått det så var man något sämre på att föra systemdokumentation bakåt i tiden.

36. X: Jo men så är det. Däremot tror jag inte det beror på åldern av systemen egentligen, utan det blir naturligt att dokumentationen blir svårare att gå igenom när man haft ett system i 20 år och det dessutom ändrats på olika sätt 80 gånger. Det kräver således en mycket djupare kunskap om verksamheten som systemet behandlar för att kunna programmera i det.

37. L: Tror du detta påverkar hanteringen av den tekniska skulden? Alltså att det hade underlättat om det fanns mer dokumentation.

38. X: Ja det tror jag verkligen. Det hade varit mycket lättare om man enkelt kunde se exakt varför något är ändrat på ett visst sätt eller varför affärslogiken är skriven som den är på ett särskilt ställe osv. Om man hade haft det dokumenterat hade det varit mycket lättare och sparat väldigt mycket tid när man ska in och göra ändringar.

39. A: Finns det något du känner att ni hade behövt förändra gällande hur ni hanterar er tekniska skuld?

40. X: Ja men det finns det alltid. Man borde prioritera det mer, man borde alltid se till att när man gör något nytt så gör man det helt rätt och dokumenterar det helt enligt regelverket osv. Sen menar jag inte att vi är dåliga på det här, jag tror många andra företag kan ha det ännu mer stressat och jag känner att vi får tid till att åtgärda och göra det så gott vi kan. Det har tex funnits väldigt mycket äldre teknisk skuld som vi känner att vi fått på rätt bana och där vi gjort det betydligt mer ordentligt. Även påtryckningar från FI om att

information ska kunna spåras och dokumenteras osv har gjort att många sådana här ändringar behövt göras.

41. L: Jobbar ni på något specifikt sätt för att prioritera skulden? Tex cost-benefit diagram osv.

42. X: Ja det gör man absolut! Tid och kostnad räknas ofta in i avvägningen och ligger dessutom ofta till grund för varför förändringar måste ske. Däremot finns det ofta många andra externa deadlines som man måste förhålla sig till vilket gör att teknisk skuld inte alltid prioriteras tillräckligt. Om man står inför valet mellan att utveckla en ny aktie robot så gör man hellre det än att ändra om i gamla funktioner tex.

43. L: Kan du väldigt grovt beskriva vad ni har för olika system och hur är de kopplade till varandra?

44. X: Det finns extremt många olika system som är skrivna i extremt många olika programmeringsspråk, tex Cobol, Java, C#, Javascript, React, Pascal. Det finns alltså väldigt många olika språk där alla ofta fyller olika funktioner. Men främst kärnbanksverksamheten är skriven i Cobol, det är alltså tex inlåning, utlåning, överföringar, bankkonton osv osv. Alltså de system som tillhör kärnverksamheten.

45. O: När är era äldsta system i från som fortfarande är i bruk?

46. X: Jag vet faktiskt inte, jag skulle gissa att några är kvar från 90-talet. Däremot kan de vara skapade då men sen har de uppdaterats 80 gånger sen dess.

47. L: Vi var inne på det innan men du menar alltså att äldre system kan bidra till större teknisk skuld?

48. X: Ja, det stämmer. Handpåläggningen tar ju längre tid oftast vilket gör att man ofta tar mer kortsiktiga beslut än att se långt fram i tiden och se vad man ska göra.

49. L: Arbetar ni med agila arbetsmetoder?

50. X: Jadå det gör vi, vi kör Scrum i vårt dagliga arbete.

51. O: Kan ni se en koppling mellan agila arbetssätt och att det underlättar växande komplexitet i system?

52. X: Ja med det skulle jag nog säga. Dels sköter man dokumentationen på ett bättre sätt då man jobbar i kortare inkrement där det är tydligt vad som ska kodgranskas och vad som ska dokumenteras osv. Sen gynnas även kunskapsöverföringen av det eftersom man jobbar i teams, man kan jobba med liknande saker och granska varandra på ett gynnsamt sätt.

53. L: Kan du se att digitaliseringen har påverkat hur ni tänker när ni utvecklar systemen? Alltså vad ni värdesätter.

54. X: Absolut! Numera är ju slutkunden i fokus medan det förr, eller ja inte ens så långt tillbaka, handlade mer om att det skulle funka för folket ute på kontoren och att de skulle kunna göra sina ärenden där. Men nu finns det mycket mer tillgång till information för personer som är kunder, och dessutom fler tjänster som är tillgängliga för kunden som inte fanns förr.

55. L: Med andra ord att kunna anpassa sig mer efter kunder och således omvärlden istället för den interna organisationen?

56. X: Ja men precis så!

57. O: Tusen tack för denna intervjun, det här var verkligen toppen för oss och vi uppskattar det otroligt mycket!

58. X: Tack själva!

7.4 Intervju 4

7.5 Transkribering av intervju genomförd 2021-05-04 X = Senior analytiker hos en av storbankerna

L = Love Anderberg

OW = Oscar Wallin

A = Axel Eriksson

1. X: Jag kan berätta vem jag är. Jag har en ganska brokig bakgrund jag gick på KTH när jag var ungefär i er ålder. Jag har några tentor kvar brukar säga. Livet innehåller en massa bananskal. Ett av de här bananskalen som jag ramlar in på var att några goda vänner till mig ja får vad som först hände det att jag skrev ett par läroböcker, programmering och nu nu finns det rätt mycket undervisningsmaterial. Men då på den tiden så var det en fullständig brist på material som inte vänder sig till till studenter på KTH eller på universitetet och det där höll jag på med sidan av att jag pluggade. Jag pluggade elektroteknik i Stockholm. Och sen, i takt med att jag höll på med de där så var det ett par kompisar så frågade de om jag inte skulle vara med och dra igång ett bolag. Som då skulle sälja personatorer och på något vis så fortsatte jag aldrig på KTH. Det var inte planen men sen hamnade jag i styrelsen i det här bolaget så klart och fatta inte speciellt mycket av balansräkningar, resultaträkningar och hur man egentligen skulle tolka. Formellt sett är jag ju liksom ekonom men liksom min skalle är ju en teknikers. Jag kunnat programmera sen när jag var liksom 16, 17 nåt sånt där och har liksom aldrig riktigt släppt tekniken även om jag inte idag sitter ju inte och programmerar.
2. O: Eller en ingenjörns gärna.
3. X: Ja faktiskt.
4. X: Så förstår ni ungefär vad är för typ av figur jag är. Och sen har jag jobbat som konsult nästan hela livet, så jag brukar säga det aldrig haft ett riktigt jobb. Jag antingen köpt in mig haha, i de bolag som har jobbat i eller också så har vi sålt bolaget och då har fått anställning haha. Jag varit anställd tror jag total 7-8 år under mina...Jag har jobbat sedan 82 så det börjar bli några år nu. Jag är 63 idag, jag vet inte tänkte nog jobba kanske 3-4 år till, tycker fruktansvärt kul att jobba fortfarande. Så jag tror att ni har satsat på rätt område. Sen har jobbat mycket med bank och finans, sedan 98 egentligen heltid. Jag var fondförvaltare mellan 2008 och 2014 och starta en hedgefond tillsammans med 2 andra killar och mitt i värsta Lehman-kraschen där. Hedgefond är väldigt mycket att liksom att bygga upp massa infrastruktur egentligen för handel, alltså ju mer automatik man kan ha desto bättre. Det bästa är ifall man kan ha en robot som handlar själv och som har själv bara sitter och kollar så att allt går bra. Vi kom väl aldrig riktigt dit, men vi hade rätt mycket som var självkörande. Vi blev tyvärr grymt osams vi 3 som utgjorde själva förvaltarteamet. Man har ju man kör inte en fond på 3 man har ju liksom risk och man har backoffice en väldigt massa människor skimrande. Vi var kanske liksom 9 personer i teamet plus alla de som säljer fonder men men själva förvaltarteamet var 3 och efter ett tag blir det ganska tråkigt när man har byggt allting som man vill bygga och dessutom så blev vi som sagt osams så.. jag började konsulta igen 2015 efter att ha gjort ett misslyckat försök att starta ett bolag som höll på med ekologiska drycker. Så jag är jag, men jag är

entreprenörshjärna egentligen. Jag inte är någon typisk storföretags kille men jag har ju jobbat på stora företag hela leva livet så att nu sedan tjugohundra femton så har jobbat som kravanalytiker, dels på *storbank där var i 3 år höll på med värdepappershandeln, framförallt räntehandel och nu på *storbank*, där jag först höll på med PSD2 i ett år ungefär och nu håller jag på med utlandsbetalningar. Utlandsbetalningar låter kanske inte så sexigt, men det är faktiskt jävligt intressant hur det där går till egentligen eftersom svenska... Jag ska inte växla in med alla de där detaljerna, men ett av problemen med utlandsbetalning är att som en svensk bank kan man inte hantera utländsk valuta. Så att så fort man ska göra en betalning i euro eller dollar då måste man ha en annan bank som är så att säga i rätt land som hanterar de pengarna. Det innebär att alla de meddelanden som behöver skicka liksom för att få de här transporter av pengar mellan bankerna, varav en hel del är bara liksom ren bokföring därför att pengarna finns någon annanstans. De är ganska avancerade och det vi gör nu det är att vi inför någonting som heter ISO20022 som i princip är som XML fast XML säger ju inget. Det är ju inte rocket science, det har ju funnits i 30 år men ISO20022 är en ohyggligt stor standard för hur man gör utlandsbetalningar. Och där liksom hela handskakningen alla liksom meddelanden som behöver kunna finnas för att liksom för man ska kunna köra flödena och helt automatiskt. De finns där med med hundratals fältet i vart ett av de här meddelandena så det är ganska stor förändring. Man kan säga att de mina 3 sista projekt har varit egentligen ganska mycket sånt som ni är ute efter. Alltså hur klarar banken och av att liksom lappa och laga sina gamla system och får det att funka i en mer modern omvärld.

5. O: Superintressant, verkligen.

6. L: Om vi skulle komma in lite på det. Hur skulle du enkelt beskriva att digitaliseringen påverkat konkurrensen på bankmarknaden det senaste året?

7. X: Ja ni ser inte jag. Det är väl ärliga svaret. Det kan ju så att säga ge en sån där varm luft som ni lika gärna kan läsa i Computer Sweden men helt ärligt är väl så här. Att man märker inte konkurrensen på så vis att alltså det är inte så att människor på bankerna går omkring och snackar om Klarna eller vilka man nu tycker är konkurrenterna. Det är liksom inte riktigt det man håller på med utan man är fullt sysselsatt med att liksom se till att ens egen infrastruktur hänger ihop och funkade så att konkurrensen är liksom mer smygande jag menar. Det är lite roligt det där, en vanlig vanföreställning, det är ju att. Banken har skit gamla system och dessutom har den dåliga programmerare och de håller på med de här grejerna sen 30 år tillbaka så fort det kommer någon som är består av unga hungriga tjugofyraåringar som just kommit ut från universitetet och kan programmera Java. Då har banken ingen chans längre. Men grejen är att det man upptäcker är att Klarna eller någon uppstickare är att det var inte så väl lätt för att... Alltså så länge som sitter i köket och utvecklar då behöver man inte bry sig om lagstiftningen eller om rapporten till finansinspektionen. Man behöver inte bry sig om någonting av det där som det egentligen innebär att vara bank, utan man kan liksom göra de lösningar som man själv tycker är bra och så långt är ganska enkelt. Men så fort man ger sig in i det här träsket som är liksom lagreglerat av av allt ifrån svenska finansinspektionen till liksom amerikanska finansinspektionen och deras regler så märker man att det har varit

komplikerat. Det har varit komplicerat på riktigt, för att det finns liksom inga tydliga regler säger t.ex flyttar det här värdet till det fältet så är vi nöjda utan att allting måste tolkas och det gör att alla de här uppstickarna måste ha en helvetes, liksom intjäningsförmåga eller ägare med stora fickor för att liksom orka blir så stora, som exempelvis Klarna har blivit. De har varit duktiga. Men, men rätt många av de här uppstickarna som säger att de ska ta över världen. De kommer aldrig komma dit därför att det är för svårt. Det spelar ingen roll att man har en modern utvecklingsmiljö. När du liksom har en kan man säga komplex och helt omodern vill jag inte kalla det för, men man har en lagstiftningsmiljö runt omkring som ställer enorma krav på liksom rapportering. Kunden vill ju veta allting som bankerna håller på med. Och någon måste göra alla de rapporterna. Se till att det flödet funkar.

- 8. O: Ja så är det. Det framgick lite när vi hade en annan dialog med med en annan bankanställd för en annan storbank och han sa just det att påtryckningar om det är lagförändringar eller med att man måste dokumentera på ett särskilt sätt som på FI vill och så vidare. Det är stora saker som bankerna arbetar med för att man ska uppfylla compliance och liknande.**
9. X: Jag har har jag kan berätta lite grann om det här MiFID 2 projektet som jag jobbar med på (En svensk storbank). Jag var där i 3 år mellan 2015 och 2018 och det gick live någon gång... 2019 tror jag. Jag får inte riktigt ihop åren nu, men det spelar ingen roll. Jag jobbade med det i 3 år i alla fall och syftet med det där var egentligen att ja det alltså... Det påstådda syftet är att man ska stärka konsumentskyddet. Man har alltid någon sån här, som låter bra i politikernas öron. Men vad det egentligen handlar om det är att man vill se till att reglerna på finansmarknaden är lika i hela Europa, så att för att kunna mäta sig mot London och New York som är liksom stora finanscentrum i världen i alla fall innan innan britterna var dumma nog att gå ur EU så... Då måste man vara ganska stor och då kan man inte ha olika nationella lagstiftningar i varje land. Så att vad man gjorde det var att man egentligen försökte införa gemensam lagstiftning för transparens för handelsregler, för rapportering och där omfattar liksom allt ifrån. Ja, hur du som kund ska veta att exempelvis att när tillgångar har gått ner mer än 10%. Ja, det är den enkla biten. Till att man reglerar telefonhandeln som sker mellan stora aktörer när man säljer räntepapper i miljardklassen och elektrifierade den för att man ska få transparens i priserna för att man ska kunna som kund se liksom bara hur de här papperna har handlats nånstans och till vilka priser? Och sen vill man att man också ska kunna se till att liksom hela marknaden får samma priser, vilket är omöjligt därför att om ni och jag köper räntepapper för vår sista miljon så är det någonting helt annat än när liksom AP fonden köper för 4 miljarder på ett snäpp. Då ger Bryssel ifrån sig, liksom en. Det är någon som räknat ut att det är 12000 sidor lagstiftning som som har kommit sammanlagt med alla uppdateringar och förändringar och då sitter vi en armé av såna här som jag och tolkar det här och försökte förstå liksom hur ska det här tolkas i vår befintliga verksamhet? Och så nämner man de här systemen köper nytt, förändrar. Och var en av bankerna har lagt ner kanske en kvarts miljard ungefär på det här. Det är ganska typiskt, liksom förändringsprojekt som är styrt av myndigheterna. Finns ingen som helst affärsnytta. Det här är ju styrt helt och hållet upp från att av Bryssel och EU. Jag tror att vi la ner kanske en kvarts miljard på det här. Och det finns 8000 banker i Europa av

varierande storlek så klart. Men det är en fruktansvärt massa pengar som bara ramlar ner i huvudet på bankerna där man kan inte göra någonting annat än att bara liksom bygga om, bygga nytt, förändra och kundnyttan är liksom dessutom ganska liten tycker jag, men det är en åsikt. Det är inte fakta men så är det. Men det där är ganska typiskt sånt som hela tiden kommer alltså mycket av det som förändras i bankerna är inte konkurrensdrivet, det är lagstiftning.

10. L: Men ändå göra med då, föränderliga omständigheter runt om. Det behöver inte vara bara konkurrensen, utan det kan vara just med nya lagkrav eller som beror på en digitalisering eller.

11. X: Ja och jag menar, alla banker försöker digitalisera på så vis att man vill ju ha mer självservice och med att kunden ska kunna göra egentligen allting själv, inklusive bolån och sånt där som ändå är hyfsat, komplicerat kan inte säga, men det är en stor affär för de flesta människor. Och drivet att digitalisera har ju funnits hela tiden. Jag tycker egentligen ganska illa om det uttrycket därför att det liksom ungefär som att det är någonting helt nytt, men datorisering har ju pågått sen jag var tonåring. Och skillnaden nu egentligen det är bara att man har i och med att man har bankid och man har liksom intelligenta terminaler i händerna på varje människa så finns det helt andra möjligheter att kunna göra saker själva och kunna identifiera sig på ett pålitligt sätt. Digitalisering så att säga utanför banken eller utanför sin verksamhet, snarare att man försöker förändra processerna inne i. Men annars är jag inte så imponerad av uttrycket digitalisering. Jag tycker att det är lite så här kejsarens nya kläder.

12. L: Om vi kommer in lite mer på just det med teknisk skuld.

13. X: Ja.

14. A: Det kan nog vara lite svårt att höra det du säger nu Love, det hackar lite.

15. X: Ja, men jag förstod ändå. Jag har fått frågorna så vet vad Love ska säga, men det hackar lite. Nej, jag kan beskriva hur vi gör rent praktiskt i det här projektet som är på nu. Jag håller på med utlandsbetalningar och vi har storleksordningen 15 system tror jag, som påverkas av att man övergår det från MT som ett gammalt. Ja, jag skulle säga att det är sjuttioalsuppfinring, eventuellt 80-talet, men det är ett ganska simpelt format där man i lite stycke text beskriver liksom varifrån pengarna ska hämtas vart de ska, vilket konto, vilka bickar som är inblandade, alltså vilka adresser och så vidare och det där har liksom vuxit över åren så att nu är liksom i formatet helt omöjligt att hantera om man ska få allting att funka STP, alltså straight through processing. Det vill säga utan manuell påverkan och problemet som vi har är att de här systemen som vi har de är, jag tror ungefär 30 år gamla. De utvecklades i början av av nittioalet ungefär. Och några köpta, men de flesta utvecklare av bankerna själva av *storbanken själva, för att han hanterat då dels kommersiella betalningar, alltså när någon av er ska betala ett hotell i Italien för ni ska förboka hotell eller vad det nu kan vara för nånting. Till att bankerna skickar pengar mellan varandra. Exempel Cobol är liksom inte riktigt kompatibelt med variabla fältlängder och dynamiska strukturer. Vad vi gör då är helt enkelt att vi gör ett och det är

nog ganska typiskt för hur man löser problemet. Att vi har ett java skal kan man säga runt omkring våra gamla legacysystem. Om vi säger att vi tar en ankommande betalning som kommer så.. Består den kanske av totalt sett 700 fält 685 av de där är rent trams. Det är ingenting som är vi är intresserade av det går att ange allt ifrån, liksom skonumret på den som skickar betalningen till liksom var vederbörande är född någonstans och vilket land och vilket datum och någon gång kommer säkert att användas. Men men det vi är intresserade av primärt, det är de 15 fälten som beskriver en betalning idag. Därför det är vad våra legacysystem hanterar. Vad vi gör då är att vi bygger en översättare som är ligger i vår moderna utvecklingsmiljö, som är java-driven. Den ligger som ett stort block framför som plockar in den där XML:en, hämtar ut de fält som är av intresse och sparar ner resten i en databas. Sparar ned vad vi fick in sparar ner också vad vi skickar ut till legacy-systemen och sen skickar vi betalningen i ett slags hybrid format där vi har plockat ut de XML-taggar som är liksom är intressanta för det mottagande systemet och det blir ju som en Cobol-koppling, och allt med allt vad det innebär i Cobol. Och det här innebär att om vi har en betalning då som exempelvis ska vända hos oss den ska liksom in i banken och ska nu ut igen därför att den skulle inte stanna hos oss. Vi är bara en intermediär. Då kan vi fiska upp de fälten som vi så att säga har filtrerat bort och slängt på vägen och se till att de går iväg i den avgående betalning som går ut igen. Och det är liksom det där gör ju det att utifrån sett så ser det ut som att vi hanterar alla de här sjuhundra taggarna, men egentligen gör vi inte det. Vi hanterar fortfarande de 15, 16, 17, 18 som vi liksom är intresserad av. Och det här systemet då, som som jag jobbar med idag, det har 5 eller 6 olika, liksom utgångar inklusive Swift gateway då som förbinder liksom banken med Swift, alltså stora betalnätet. Och sen får vi betalningar, dels från några hundra banker runt om i världen plus 2 stora clearinghus som hanterar hanterar euro och det där. Det där är ju ett sätt att förlänga byggt på längden hos våra cobol system med med 5 till 10 år kanske. Parallellt med det här så pågår någonting som heter p27 som ni kanske hört talas om. Ja, det har varit jävligt hemligt fram till nu. Nu börjar jag vara ganska publik, men bankgiro, som också är en gammal 80-talsuppfinning vill också gå över till *ohörbart*, alltså hela världen på betalsidan går över till XML, det är 20022 betyder. Och då bygger man helt ny infrastruktur som ersätter gamla bankgirot. Man bygger helt nya betalsystem och där har man valt alltså en annan approach än den som vi har på den del som jag håller på med. Vi bygger liksom en rapper runt omkring en slags ja... Vi skyddar de gamla systemen mot den besvärliga omvärlden, både för både för inkommande och utgående betalningar, men däremot va mina kollegor som är är kanske 40 pers någonting som jobbar med det här, de bygger ju helt nytt och sen ansluter dem till den gamla infrastrukturen bara liksom på själva utgången. Det behövs lite kopplingar till de gamla systemen också. Så att inom banken finns det egentligen 2 helt olika sätt att hantera liksom den här, ja, tekniska skulden.

16. L: Men, för du berätta ju lite om de här strategierna ni har då, om jag förstår det rätt så är det att ni att man liksom bygger runt det så att säga, bygger runt den gamla legacyn.

17. X: Ja.

18. L: Men kan du se att det finns en teknisk skuld kvar hos den gamla legacyn och att den har negativ påverkan på något sätt? När ni bygger runt om så att säga, skulle du hellre bygga ett helt nytt, eller?

19. X: Ja, men det är så här att problemet med att bygga helt nytt är att det finns... alla de här gamla systemen innehåller liksom verksamhetsregler som har vuxit fast och vuxit fram under kanske 30 års tid och vi hanterar ju dels *storbank* sen har vi alla enheter runt om i världen, alltså till Tyskland, Norge, Finland och så vidare och om betalningar... ehm.. Betalningar som ska ut från Sverige eller så här: Alla banker har ju kontakt med andra banker som som håller liksom utländsk valuta åt dem. Vi har exempelvis *ohörbart* som är vår dollar bank. *Storbank* kan liksom inte hantera dollar därför att vi finns ju i Sverige. Vi har svenska kronor. Man håller ju inte på att växla och greja liksom, utan man måste ju ha en bank som finns i rätt land och som är liksom vår bank som hanterar våra dollar. Och på det viset har man då korrespondentbanker i alla valutor som som bankerna måste kunna hantera. Jag tror att *Storbank* kanske hanterar 50 valutor. Jag bara gissar, jag har ingen aning. Jag har en lista över dom men jag inte räknat och det innebär att man då måste ha 50 banker som hanterar de här valutorna allt ifrån turkiska sån här vad heter de? Ja, deras funny money till liksom dollar då som är den största valutan. Och om då de finska betalningar exempelvis, om de ska skicka turkiska vad dom nu heter, fan att jag inte kommer ihåg det. Säg japanska igen då. Då har ju inte de direktkontakt med den vanliga yenbanken utan det har ju Sverige utan då skickar de betalning till oss och så och bokför vi det på deras konto och så skickar vi betalningen som ser nästan likadan ut från *Storbank* till liksom yenbanken och sen får de skicka vidare så att pengarna så småningom når adressaten som kanske inte ens finns i Japan. Det är kanske så att vi gjort upp med någon i Singapore att de ska ha betalt igen, alltså sånt som är fullt möjligt och då är det liksom en lång kedja banker som är inblandade. Och det gör ju det att alla dom här verksamhetsreglerna för hur man ska kunna hantera vad som händer om finnarna skickar en betalning i valutan X, dem finns ju inbakade i de här gamla systemen. Så att ja, det är en teknisk skuld, men det är också liksom verksamhetsregler som liksom ingen jävel har full koll på. Och att om man då skulle säga så här: men vi byter ut alltihop och skriver i java, för det har vi läst att det är så mycket bättre, vad som händer då är att det är att de är ett gigantiskt kravarbete framför sig att liksom faktiskt beskriva vad alla de här gamla systemen gör och det där är väldigt ändringskänsligt.

20. L: Med andra ord så om man skulle byta ut, lite som du säger, om bankerna bara skulle programmerar om allt i java. Då uppkommer en helt ny teknisk skuld?

21. X: Ja och skulden är ju inte riktigt tekniskt då, utan det handlar ju om att man måste på något vis sätta ner kraven på vad det egentligen är som det gamla systemet gör i detalj. Och nu har ju bankerna hyfsat bra koll på allt sånt där, det finns ju värre verksamheter där man faktiskt inte vet riktigt vad de gamla systemen gör. Jag har jobbat på sådana också, men det är inte det ni frågar om, men det där är så pass stor sak att göra i en verksamhet som funkar och funkar varje dag med miljardbelopp som åker in och ut så att man vill liksom inte röra det där man inte måste. Alltså, vi har ju tusentals betalningar som går i de här systemen hela tiden och om vi missar en finansiell betalning till exempel, alltså

betalning till en annan bank och inte får iväg det därför att vi har liksom driftstopp just nu i vårt system. Då kostar det böter, för de andra bankerna har ju inte pengar på kontot. Då måste de låna ut till oss och det gör de ju inte utan att de liksom tar betalt för det. Så det blir ett jävligt dyrt att göra fel. Gör man fel i ett system som hanterar kommersiella betalningar kanske man har 1000 betalningar som det är något fel på. Men vem ska rätta alla dem? Det är manuellt arbete, man kan inte rekrytera 1000 indier som tar en betalning var utan det är liksom en befintlig organisation som måste fixa det och det där är en mardröm. Så att därför vill man göra förändringarna i små steg som är kontrollerade hela tiden.

22. L: Prioriterar ni olika typer av teknisk skuld och har ni någon speciell strategi för hur ni prioriterar?

23. X: Ja, jag beskrev tidigare att vi jobbar då både med p27 som är liksom ersättaren till bankgirot. Det här 27 är 27 miljoner invånare som kommer att omfattas av det, så det är ett samarbete mellan de nordiska länderna. Norrbyggarna är inte med av någon anledning, men det är Sverige, Danmark, Finland. Jag får inte ihop 27 miljoner, men det spelar ingen roll.

24. O: Det ska göra det enklare att skicka pengar mellan de nordiska länderna om jag förstått rätt?

25. X: Ja, det kan man säga och du som kund kommer inte märka någonting av det här. Du skickar fortfarande hyran till ett bankgiro och skriver att det är sjutusen spänn som ska betalas in. Men bakom det här så är liksom en helt ny infrastruktur som erbjuder att man skickar... Ja alltså det man vill i framtiden alltså framtiden är ju fan nästa år nästan men det är ju att ekonomisystemet ska ju kunna skicka en klumpbetalning från en kund på säg 7 miljoner spänn från ett stort företag och skickas med en enda betalning och i betalningsinformationen som då är ett XML-paket istället för liksom en textsnuttt i gammal standard. Nu ska ju också ekonomisystemet kunna berätta att det är de här 27 fakturorna som är betalda. Det här beloppet på varje faktura så här mycket moms är det den här fakturan har kvittat mot den här kreditfakturan som som låg inne och ja, alltså den typen av information som gör att man kan automatboka. Det här är ju inte som banker tjäna pengar på. Där får man ju bygga affärstjänster runt omkring som gör att man faktiskt får tillbaka de här pengarna. Men den här nya infrastrukturen som vi bygger den kommer ju tillåta den typen av liksom nya tjänster som så småningom även ganska små företag kommer skrika efter. Och vad vi gör väldigt tydligt är att vi behåller livslängden på de gamla systemen några år till. Några år är kanske realistiskt mer än 5, men inte inte 10. Genom att lappa och laga och bygga den här liksom tekniska rappen. Samtidigt så överför vi mer och mer funktionalitet till liksom den infrastruktur som man bygger runt p27 så att det är java system som man bygger mycket av p27. Det blir liksom det new kid on the block som ska ta över mycket av funktionerna i de gamla, men det blir liksom en stegvis väldigt kontrollerad övergång och ingen pratar om att man ska ersätta de gamla systemen liksom redan nu därför vi har fullt upp med den här migreringen som vi gör men det är det som kommer hända att man flyttar över funktionalitet och börjar

med enkla systemen och sen tar man det som har mycket transaktioner och stor kapacitet sist.

26. O: Så men man kan väl se det nästan som att man rullar ut det successivt för att få ut det helt enkelt?

27. X: Ja alltså banker är ganska försiktiga organisationer och man vill ha koll på det man gör och det har man ju oftast. Det är väldigt sällan som det uppstår fel i banksystem faktiskt. Även om Aftonbladet och Expressen har helsidesuppslag när det händer någonting. Men det är väldigt sällan som det sker såna saker som att det kommer dubbla betalningar eller liksom att det saknas pengar på kontot fastän jag borde ha en miljon stående där.

28. L: Nu har du jobbat på 2 storbanker som är väldigt traditionella i sitt sätt, lite som vi pratar om med bolån och man har ju flera avdelningar och det är personlig kontakt och så vidare, ofta när man jobbar mot kunderna....

29. X: Nja inte riktigt alltså. Du har personlig kontakt om du ska låna pengar till en lägenhet eller en bostad. Då är det fortfarande många människor som vill prata med en riktig handläggare, men... Jag vet inte.. nu kanske jag inte är typiskt för att vara bankkund, men alltså även *storbank* som har varit tjurigast med att behålla sitt kontorsnät, lägger ju ner en del av det därför att kunderna kommer inte in helt enkelt. Man sköter allting via mobilen och datorn och paddan. Så att det pågår ju en jätte förändring av, liksom kundbeteendet också och det beror ju på att om man räknar lite bakåt, min svärfar som tyvärr inte i livet längre. Men han var liksom den sista generationen som inte hade haft dator på jobbet. Han gick i pension han var 55 år han var född 26. Han slutar jobba i början av åttiotalet och IBM PC kom nittonhundraåttiofyra, så alla som som jobbar 84 eller senare har man ju liksom haft dator och det är ju det att tröskeln för att kunna sköta en padda eller en smartphone, den är ganska liten. Och dem här 90-åringarna som kräver att få stå och köa utanför en postlåda liksom, de de dör ju av hundra tusen varje år så att det blir färre och färre som inte klarar den här digitaliseringen. Jag avbröt din fråga, men berätta din fråga.

30. L: Nej, det var ingen fara, men alltså det var ju jag som egentligen utvecklade lite fel. Frågan var egentligen bara om du kan se att tidspress från andra avdelningar på kontoret leder till en ökad teknisk skuld, alltså att ni skyndar på erat utvecklingsarbete?

31. X: Nej. Nej egentligen inte. Alltså det vi får in som utvecklingskrav. ...alltså jag jobbar i stort agilt tåg med med hundrafyrtio pers ungefär. Vi har 13, 14 smågrupper. Vi är väl 9 stycken i vår, inklusive vår scrum master och det vi håller på med det är ju nyutveckling som kunder inte ser, men vi har ju andra tågagnar om man säger så som jobbar med grejer som är mer kontorsnära. Ofta är det ju så att det som kontoren vill för att det förenklar deras verksamheter ska ju in i den stora prioriteringslistan, liksom så att ...Det är inte så att att ja, de får inte igenom allting som de vill ha, utan de får ju ofta vänta på grejerna så kontoren har en massa manuellt arbete som de hoppas slippa liksom och som inte blir gjorda bara för det finns annat som är mer viktigt och mer bråttom.

- 32. O: För det vi liksom hört tidigare, det är att oftast hos mer traditionella aktörer att det finns en del avdelningar som inte är särskilt tekniskt lagda i och med att kärnverksamheten faktiskt inte har varit digital från början och att det därmed på grund av den hur ska man säga... Obalansen i kompetens att det kan bli en del teknisk skuld som uppstår just på grund av det för att en mindre tekniskt lagd avdelning kanske inte förstår hur tidskrävande det är eller komplexiteten i att ändra någonting litet att det tar inte lång tid att titta på det, men däremot att det ska gå igenom kodgranskning och det ska gå igenom den substansen osv att det på sätt och vis omedvetet uppstår en teknisk skuld omkring det så att säga. Men det upplever inte du?**
33. X: Nej, jag, jag förstår precis vad du menar, men jag känner inte igen mig. Men det är möjligt att du skulle få ett annat svar av någon som jobbar med andra bitar i banken. Men däremot är det ju så att alltså, det här är ju ett maskineri, en bank eller vilket företag som helst är ett maskineri som man inte gärna ändrar i. Därför att om det fungerar, if it is not broke don't fix it, och så fort man försöker förändra någonting då är ju risken också att man förändrar flöden som funkar och att då fastna i någon rapportering eller man behövde ha det här värdet till någonting som ingen fattar just då att man behövde ha därför att det man, den personen som visste det eller den avdelning som visste det, de var inte tillfrågade så att det finns ju... det är klart att det finns en tröghet i att göra förändringar som man inte ska underskatta.
- 34. L: Övergripande Ingmar, finns det någonting du känner att ni borde ändra på när det gäller hanteringen av teknisk skuld?**
35. X: Ja alltså vi har en sak som egentligen inte är teknisk skuld men som är onödigt komplicerade hos oss och det tror jag gäller de flesta banker. Infrastrukturen är liksom riggad för att vara så säker som möjligt. Vi har ju brandväggar även innanför liksom brandväggen om man säger så. På *storbank* hade man inte det utan då hade man liksom ett skyddslager och sen om man vill ha öppningar i brandväggen så var det nånting som man behövde beställa lång tid i förväg. Här har vi liksom multipla brandväggar även innanför så att det är mer en slags ringmekanism och det här gör ju att ska man beställa servrar eller beställa liksom databaser eller den typen av grejer så blir det ofta ganska byråkratiskt och det gör att saker som egentligen inte borde ta någon tid tar väldigt mycket tid och det där är inte riktigt det ni frågar om. Det är inte riktigt tekniskt guld, men det är samtidigt så att det problemet skulle aldrig en startup ha alltså att det tar tid. Vi väntar 3 månader på att få en server. Ja, men ni vet lika bra som jag att en server är ju liksom en bit av en hårdvara sitter i skåp. Det är inget komplicerat, det bara trycka på en knapp så är det klart. Med min gamla verksamhet så hade vi skaffat en Amazon server på på liksom en timme inklusive liksom fikarasten också. Här tar det liksom en oändlig tid därför att det är så mycket jox runt omkring som ska formuleras, det här är ett *ohörbart*, så jag vet att banken har ju liksom fixat den här beställningsmekanismen nu på sistone bara några veckor sen så att det är otroligt mycket enklare idag. Det är inte riktigt den typen av problem som är teknisk skuld, men det är ju fortfarande så att all den här säkerheten som bankerna har den kostar ju och det är ju en av kostnaderna att vi sitter

med grejer som har utvecklats och som bara går att köra på en lokal maskin liksom därför att vi har ingen applikationsserver.

36. O: Det kan jag ju säga leda till teknisk skuld. Det är inte i sig teknisk skuld, men det kanske kan leda till teknisk skuld i längden att man har den typen av arbetssätt eller att det sätter lite käppar i hjulen.

37. X: Ja. Ett annat problem som vi har som inte heller riktigt har med teknisk skuld att göra. Men det är om vi vill plocka in open source programvara till exempel vilket det finns rätt bra verktyg för transformation av XML att mappa från ett format till ett annat ganska automagiskt. Då vill man kunna plocka open source programvara som är allmänt beprövad och känd. Problemet är att då måste någon bli systemägare för den programvaran och liksom garantera att man har support för resten av banken. Alltså så här, om ni kommer in på storbanken och så säger ni så här, men fan det här kan vi använda *ohörbart* för. Den utåt mot en massa utvecklare på banken som ska se till att man får svar på frågor att man kan garantera driftstabilitet, att det finns en supportorganisation som ser till att den här snurrar tjugofyrasju och så vidare. Det är så här man behöver inte vara, eller man måste nog vara ganska, ska jag säga luttrad för att säga "nej jag tar den risken". Nu kör vi liksom så att.

38. A: Ja, någon måste liksom äga processen.

39. X: Ja och även om det är lätt att förstå drivkrafterna bakom att man ska göra så, så blir det väldigt dysfunktionellt för organisationer att man inte tar in nya saker därför att det blir för administrativt jobbigt för dem som i så fall på den här bördan, och det är inte riktigt tekniskt guld, men det är ju ganska närbesläktat.

40. O: Ja, verkligen. Du nämnde tidigare att ni arbetar agilt och då har vi väl egentligen en fråga som kanske är lite svår att koppla men frågan lyder om du kan se en koppling mellan det agila arbetssättet och att det faktiskt underlättar hanteringen av växande komplexitet i systemen? För att förklara lite tydligare. Vi tänker då att i och med att man arbetar lite mer fram och tillbaka, att man har en lite mer flexibel arbetsstruktur och att det på sätt och vis underlättar att man hela tiden kan gå tillbaka och göra om göra rätt så att säga.

41. X: Ja, jag förstår precis frågan men jag tror inte riktigt jag har svaret. Men om jag får prata en stund runt det med risk för att jag blir lite oorganiserad så. Jag jobbar i 2 team i praktiken, jag är knuten till ett team som består av Java-programmerare. Vi är nio i teamet inklusive mig som inte utvecklar utan är kravare, och så har vi en testare. Sen är vårt grannteam Cobol-utvecklare i varierande åldrar, lite äldre än vad ni är men inte jättegamla, och de har plockat in 2 duktiga trainees som är systemvetare och som har fått internutbildning i Cobol därför att det är brist på Cobol-programmerare. De flesta är min ålder och har eller ska snart gå i pension så att det är självklart en bristvara på marknaden, samtidigt som det finns en enorm kodbas som kräver att man faktiskt kan ändra i systemen och veta vad man håller på med. OK, det här agila då? Då är det så att de här Cobol-programmerarna då, de är ju uppvuxna med det här att först skriva en

kravspecifikation och sen är den är bra och den är perfekt, då utvecklar man det och sen när man har utvecklats färdigt, då testar man. Sen hittar man buggar och sen så är det klart och då sätter man det i produktion. Problemet med det här, något ni vet mycket väl, det är ju att det hinner gå 2 år innan man är färdig och under den tiden så har något nytt hänt som gör att kravspecifikationerna inte stämmer riktigt, och det är inte jättelätt att få en del av de här människorna att jobba agilt därför att de vill inte riktigt det. En del av detta är självklart dumheter, att man inte är van. Men en del av det är också högst rationellt eftersom att det som händer när man arbetar agilt är att man har självstyrande grupper där ingen riktigt tar ansvar för dokumentation och förvaltning, för grejerna måste ju leva framöver. Utan i en perfekt agil värld hoppar det team som är mest sugen på uppgiften in och löser uppgiften och sen dyker man in i nästa. I praktiken är det istället så att man måste ha ett system som man är knuten till, så att man har expertkompetens om just det här systemet. Och jag tycker faktiskt det kan vara svårt att få ihop den här agila, lätttrörliga världen med att man faktiskt måste ha långsiktig stabilitet och människor som faktiskt vet hur grejerna sitter ihop, och även fungerande dokumentation osv. Alltså agilt arbete har ju kommit fram i en miljö som inte är inom banken om man säger så, en miljö som alltså inte är tyngd av finansinspektionen eller där kunderna liksom flyr till en annan bank om det blir fel någonstans. Utan det är lite mer så här att, vi har 2 veckor på oss att se till att vi tex får ett nytt fält för sökningarna, ja då petar vi in det så har vi en demo där vi visar hur det funkar och så säger användaren att det ser skitbra ut, och så deployar man det. Men bank är något helt annat. Det händer väl i alla branscher att de säger så här, vår bransch är så jävla speciell och ingen annan förstår hur vi har det. Men bank är ganska speciellt. Jag har ju varit konsult i hela mitt liv och det här att man är så regelstyrd är annorlunda i bankvärlden och därför krävs det ganska noggranna analyser innan man sätter igång och ändra på saker. Det är komplexa flöden och många system som är inblandade. Därför krävs förstudier som är så långt ifrån agil utveckling man kan komma. Jag tror att egentligen att det som krävs och vad vi behöver jobba med, det är en slags blandning av gammalt Vattenfalls tänk i de tidiga faserna, och sen när man kommer in på själva implementationen så är det mer naturligt att jobba agilt tror jag.

42. O: Jag tänkte på just den här komplexiteten i och med att programmeringsspråken, kobolt till exempel, då är äldre vilket innebär att det kanske inte riktigt finns lika mycket dokumentation och information om det. Det kan vara som du förklarade tidigare att systemen faktiskt är väldigt gamla, hur fungerar det då med dokumentationen. Alltså är det en problematik som man stöter på ofta, just att det är svårt att finna tillräckligt med dokumentation och att det därmed kan ge negativa konsekvenser gällande den tekniska skulden och att man helt enkelt hamnar efter på grund av det?

43. X: Nej, det är den ju inte riktigt. Egentligen har jag inte superbra koll på hur cobol-systemen är dokumenterade. Jag arbetar ju inte direkt i de utan jag jobbar tillsammans med dem som håller på med dem, vilket innebär att det jag behöver ta reda på är muntlig tradition. Men jag skulle säga det att hur mycket man dokumenterar och hur bra man dokumenterar, det är ganska personberoende. Jag har tex läst fantastiskt dokumentation om ett av våra system som jag jobbat med, som var som ett skolboksexempel på hur dokumentation ska se ut. Rätt mycket annat är däremot illa dokumenterat och sitter i

huvudet på dem som har jobbat med det, så jag tror att det är en rätt stor blandning och att det skiljer sig väldigt mycket.

44. O: Vi har hört att det inte är riktigt lika mycket open source känsla kring just de äldre programmeringsspråken, det kan tex vara svårare att hitta information på nätet osv. Hur ser du på det här och hur påverkar det egentligen?

45. X: Nej, det är klart att även utvecklingsmiljöerna för cobol också har förändrats och uppdaterats. Men när vi utvecklar java-systemen, då gör vi exempelvis så att vi börjar med att ta fram ett testfall och så bygger vi koden runt testfallet. Sen kör vi automatiska tester som går varje morgon eller natt och spricker någon av dem, ja då vet vi att vi har gjort en förändring som gör att våra gamla testfall inte fungerar. Sen ibland så är det ju fel vi upptäcker, tex att vi har kravat fel eller liknande och då får vi förändra testfallen. Men den där möjligheten att direkt i koden göra enhetstester och automatstester finns liksom inte riktigt i Cobol-världen, vilket kan göra det mer problematiskt.

46. L: Kan du se att utvecklingen på marknaden har påverkat vad ni värdesätter när ni utvecklar systemen?

47. X: Jag kan inte svara på det och det har att göra med att jag har jobbat med PSD2 tidigare. Alltså det här öppenhetsdirektivet som gör att man egentligen kan gå förbi bankens gränssnitt och kör från sin egen mobil. Eller ja man kan ju tolka det på olika sätt men de delar av banken som märker av den här förändringen ordentligt, alltså det som är kantat av digitaliseringen, det är de delar som håller på med mer konsumentnära saker. Det kan tex handla om tjänster för att digitalisera bolån, vilket har varit en central grej på de storbankerna jag jobbat. Med andra ord att i princip kunna köra en hel bolåneaffär från mobilen, utan att behöva komma in på kontoret om man inte vill det. Och de delar av verksamheten märker mer sånt där, alltså det som ni frågar om. Jag ser inte riktigt det där på samma sätt beroende på att jag jobbar med fel, eller rättare sagt andra saker.

48. O: Då kan vi faktiskt komma in på det här med PSD2, eftersom att det är en del av vårt uppsatt. Så hur ser du på det? Anser du att PSD2 och Open banking på sätt och vis har förändrat förutsättningarna rent konkurrensmässigt på marknaden eller vad skulle du säga har förändrats?

49. X: Ja nu lämnade jag PSD2 precis innan det var sjösatt och färdigt så att säga. Men jag har ju massa erfarenheter från hur vi byggde det där och och hur svårt det var. Men däremot vet jag inte om det påverkat banken framöver och hur man gör. PSD2 har egentligen inneburit en jättekoalition mellan bankens traditionella säkerhetssystem och liksom OAuth 2.0 som är token baserat. Vad vad man gjorde var att alltså alla banker har ju säkerhetslösningar som innebär att man spärrar vilka kunder och vilka användare som ska få tillgång till vad. Rent praktiskt så innebär det att även om du ser i internetbanken att det finns möjlighet att tanka ner en betalbil exempelvis så kommer du inte åt det. Det är liksom gråmarkerat eller så syns det inte överhuvudtaget för att du inte köpt den tjänsten. Det här är ett centralt system som håller ordning på vad en viss användare får och inte får göra både innanför banken och utanför, och det man gjorde för att kunna

klara av PSD2 var att tillåta TPPs, alltså tredjepartsleverantörer, att få kunna ta del av informationen. För att få vara TPP måste man vara certifierad och för att få certifikatet från FI måste man nästan ha ställt sig samma krav som för att starta en bank, alltifrån regler för aktiekapital till att styrelsen är oförvitlig och inte icke brottsdömd osv. Men så länge man har certifikatet har ingen bank rätt att neka dig att få göra det du vill, utan antingen signar man upp sig med certifikatet eller så skickar man in det vilket gör att säkerhetssystemet räknar in att den här aktörer har rätt att kolla på kontona för våra kunder eller vad det nu kan va. Sen måste kunden i sig naturligtvis ge access med hjälp av exempelvis bankid eller någonting. Hela den här lösningen var någonting som bankerna byggde ovanpå sina befintliga säkerhetssystem så att i princip så är PSD2-access en ny produkt i produktkatalogen för säkerhetssystemet om man säger så. Det är så man hanterar det, alltså har man rätt att titta på konton, då är det en ny produkt som just den här TPPn har och TPPn har ett kundnummer som vilken kund som helst. Sen byggde man logik för att hantera hela den här refresh av token osv i java. Så där byggde man ett nytt skal utanpå de gamla banksystemen som gör själva handskakning och säkerhetshanteringen. Sen vad banksystemen, eller de som numera kallas legacysystemen gör, det är att de vaskar fram alla de här transaktionerna om man vill titta på dem, eller att man skickar in ett meddelande för att starta en betalning till de banksystem som sköter utlandsbetalningar eller vad det är för typ av betalningar. Så att där har man byggt en lösning som är ganska lik den jag håller på med som är som ett skal som ligger mellan de gamla banksystemen och omvärlden. Men har det här med PSD2 påverkat bankernas digitalisering särskilt mycket i övrigt, nej egentligen inte. Däremot tror jag att det är möjligt att vi för en större påverkan på sikt, där folk kanske väljer TPPs betalningslösningar och för att titta på konton osv, istället för att köra internetbankens telefon GUI. Men jag tror att den skillnaden nog kommer att synas mer på sikt. Alltså att det kommer vara en glidande övergång från bankernas gränssnitt till de mer generiska, och det är klart att det kommer öka konkurrensen eftersom att om bankerna bara är en tjänsteleverantör till din app du har i mobilen så har man kopplat bort en del av kundtroheten. Det är liksom mobilen som blir tittfönstret in till den leverantören och just nu kanske det råkar vara Swedbank men imorgon är det kanske någon annan. Plus att man förmodligen redan har pengar på mer än ett ställe. Man sköter sina betalningar och sin lön på ett ställe medan man har sitt sparande på ett annat osv, så på sikt kommer det här självklart att öka konkurrensen alldeles oerhört. Men vi är för tidigt ute för att kunna säga exakt vad som har hänt och hur det än så länge påverkat, vi är inte riktigt där än.

50. O: Det här med mikrotjänster arkitektur har blivit ganska centralt efter PSD2 och open banking i allmänhet, är det någonting du ser eller vad är din tanke om det?

51. X: Jag är fel person att svara på frågan för att jag är inte arkitekt, utan jag är kravare och även om jag tänker som en arkitekt så är det inte riktigt mitt arbetsområde att fundera över den typen av grejer. Men det är klart att det pratas om mikrotjänster i alla banker och har blivit alltmer centralt.

52. O: Tusen tack för detta samtal, oerhört intressant att få lyssna på dig och ta del av ert arbete!

7.6 Intervju 5

Transkribering av intervju genomförd 2021-05-04

L = Lennart Lindholm (CISO på Carnegie)

Lo = Love Anderberg

O = Oscar Wallin

A = Axel Eriksson

1. **O: Vi kan väl börja med vad du heter och vad din roll och arbetsuppgift är på Carnegie?**
2. L: Yes, Lennart Lindholm heter jag. Jag är CISO eller Chief Information Security Officer då på Carnegie och har varit här i ungefär 4 år, så jag är ansvarig för all vår IT- och informationssäkerhet i hela Carnegie Group. Vi finns ju inte bara i Sverige utan vi finns även i andra länder, så att i min roll är en global roll. Jag ser till att vi har en tillräckligt bra säkerhet.
3. **O: Och vad har det för arbetsbakgrund innan Carnegie då?**
4. Ja, jag har jobbat som CISO i kanske, jag vet inte längre det är nu, 12 år kanske. Jag har haft den olika CISO-roller på andra organisationer tidigare också och innan dess jag jobbar ju som typ IT-chef då eller ansvarig för IT-miljö, så att jag har ju jobbat med IT-frågor typ sen nån gång i början eller mitten av nittioalet.
5. **Lo: Men om vi kommer in lite på banksektorn. Om du kan svara på dessa frågor. Hur skulle du att beskriva att digitaliseringen har påverkat konkurrensen på bankmarknaden de senaste åren?**
6. L: Jag skulle nog säga att det har varit en positiv utveckling, att de har drivit bankerna åt det här hållet. När regelverket dök upp första gången, då med psd2, så vart det ju lite... Kollegorna i branschen kanske tyckte det var lite läskigt eftersom man måste ju släppa in andra aktörer och låta dessa ta del av bankens tjänster och konton och så vidare. Men jag tror att det bara har blivit bra för att öppna upp för marknaden och jag tror med backspegeln, eller om man tittar bakåt så har det inte försämrats för några banker utan snarare har det egentligen öppnat upp möjligheter för att ändra sina tjänster eller mer att hänga med i digitaliseringen. Man var ju rädd lite grann som storbank att nischbanker skulle gå in och och ta marknadsandelar eller lansera tjänster och ta kunder från storbankerna. Men jag tror storbankerna har vänt på det och sagt att det här kan ju vara till vår fördel också. Att har vi bra tjänster så kanske kunderna stannar kvar också, även fast det finns andra nischbanker som nyttjar det här. Så att min spaning efter ett antal år sen första utkastet av psd2 kom, det är ganska många år sedan nu, så har det ju svängt från att vara oro till att det har faktiskt blivit något positivt, så så tycker jag.
7. **O: Ja, då var vi ju lite inne på konkurrensförändringarna. Men hur skulle du säga då för dig eller för er hos Carnegie, har det varit positiva effekter eller har det varit negativa effekter?**

8. L: Ja alltså, nu är Carnegie ingen storbank så ska vi vara krassa så träffar inte regelverket 100% så att säga. Det är tvingande för den stora bankerna, men vi ligger precis på gränsen under så att säga. Om man nu räknar med de stora svenska bankerna då. Så vi är lite mer nischade så att vi... eh.. jag vet inte om det har drabbat Carnegie speciellt mycket på det sättet heller och det är inte. Det är ganska få spelare som vi konkurrerar med egentligen i det stora hela.
- 9. O: Ja ni rör ju ett annat område så att säga.**
10. L: Ja precis så jag upplever inte att vi har sett någon ökad konkurrens på det sättet mer än att vi själva känner att vi, men det är ju allmänt i samhället att man gör en digital förflyttning precis som alla andra företag, så att det är ju att det kanske aktualiserar frågan lite mera men snarare så att det har liksom aktualiserat frågan och gjort att vi måste titta på lite mera.
- 11. O: Ja, men vi kan väl egentligen gå vidare åt det mer tekniska aspekter och prata lite om teknisk skuld, så då kan man väl börja med om du känner du till begreppet?**
12. L: Det tror jag, men om du har en annan definition så får du kanske dra den då.
- 13. Lo: Det innefattar väldigt mycket, så vad vi tänker på specifikt, det kan vara flera andra grejer också, men det här med att man tar genvägar på grund av tidspress som gör att du sätter system i skuld som någon gång måste åtgärdas. Men det kan ju också bero på personalomsättning och allt vad det kan vara. Men vi utgår egentligen mer utifrån vad du, vad du, vad du ser teknisk skuld som.**
14. L: Ja, det är ju lite ett buzzword egentligen, eller legacy så man brukar kalla det för också. Så att det är ju ett buzzword som, jag vet inte när det dök upp, men för ett antal år sedan och då var det liksom att vi har så mycket legacy som vi måste hantera.
- 15. Lo: Det för in oss lite på nästa fråga då, så skulle du säga att er verksamhet är påverkad av teknisk skuld?**
16. L: Alltså, för mig är ju, jag jobbar ju med säkerhet, så för mig är det ju att se till att våra system är uppdaterade så att säga och att det fortfarande finns support att de lever och att vi kan uppdatera systemen, både hårdvaru och mjukvarumässigt. Så att det finns ju end of life på många ställen så att vi ser ju till att ha en life cycle management som gör att det här med att ha Windows då att vi som kör Windows server 2008 eller någonting, det är end of life och så vidare, nu är det Windows 2012 och det går ju end of life om ett par år så att vi måste se till att har vi några system som kommer bli end of life så småningom då måste vi börja i god tid för att byta ut dem eller uppgradera dem för det är ju en tröghet i affärssystem och så vidare att byta ut såna där saker. Det är en aspekt när man tittar på säkerhet så den måste man hela tiden se till att uppdatera både hård och mjukvara, men sen finns det ju den andra delen som vi var inne på med kärnsystem som man har liksom haft väldigt länge och som man har byggt på och byggt på. Återigen Carnegie är ju ganska ungt bolag så vi har ju inte funnits i såg 50 år utan våra system är hyfsat moderna

om man kallar det för det. Så att vi tittar ju på det där men ur vår aspekt så är den strategi som vi har när gäller digitalisering kräver att vi byter ut en hel del av vår grundprogramvara så att säga, våra kärnsystem. Så att utifrån den aspekten så har vi ett just nu ett stort projekt igång för att just byta ut våra nyckelkomponenter så att vi så har ju tagit ett litet större grepp om att byta ut oss så att säga. Alltså byta ut just delar av våra kärnsystem som inte är framtidssäkra om man säger så utifrån digitaliseringsperspektiv, utan vi kräver system som är mera flexibla och snabba och mera föränderliga och det har vi inte riktigt i våra nuvarande system då.

17. Lo: Skulle du säga att det är på grund av av teknisk skuld som gör att de inte är så lättföränderliga och anpassningsbara?

18. L: Nej, det handlar ju mer om marknaden, alltså vart vart vill vi ta vägen? Vad vad ser vi om vi tittar liksom framåt i kristallkulan, vad behöver vi befinna oss om om 5 år eller 10 år eller kanske 3 år om vi ska vara ett fortsatt lönsamt företag? Så att gamla meriter kan man ju inte stå på utan du behöver hela tiden kunna förnya dig och erbjuda tjänster till dina kunder. Så att det här är ju mer ett affärsmässigt strategiskt beslut som vi har tagit. Sen kan man ju säga att det har kanske med skuld att göra lite grann eftersom vi har ju byggt in oss lite grann i det nuvarande systemet då men så det är en beroende på definitionsfråga hur man ser det.

19. O: Så det handlar egentligen om att ni anpassat er efter omständigheterna och anpassat er efter digitaliseringen och därmed gör förändringar för att helt enkelt vara helt up to date?

20. L: Ja i vår situation skulle jag säga att det är applicerbart för Carnegie, ja. Vi har inte några stordatorer som står liksom och tuggar i någon källare sen 20 år tillbaka, vi är för unga för det så att säga rent systemmässigt. Så vi kanske inte har den skulden på det sättet att vi helt plötsligt nått vår livs ände utan det handlar mer om strategier att vi för att kunna ta klivet i fullt ut mot digitalisering så behöver vi ju förändra vår grundstruktur.

21. Lo: Du sa någonting där med någon Windows life cycle, är det någon speciell strategi ni har för att motverka tekniskt skuld?

22. L: Nja, det handlar om life cycle management och jag tog Microsoft som ett exempel. De har ju ganska tydlig så att säga deras support cyklar är ganska tydliga. Jag menar Windows 7 var ju ute 2008 osv och sen kommer ju Windows 10 år 2012 och allt det där. Och eftersom vi är ganska Microsoft tunga i vår IT-miljö så gäller det att vi måste ju följa Microsofts supportcyklar också och se till att vi inte hamnar efter.

23. A: Du menar alltså så att ni är kompatibla med deras system helt enkelt?

24. L: Ja, för de slutar ju då tex att skicka ut säkerhetsuppdateringar efter ett antal generationer bakåt, och där är det ju mitt ansvar som IT säkerhet att säga att vi inte ska ha några system som inte är supportade. Vi måste kunna få säkerhetsuppdateringar hela tiden på alla system vi har. Det gäller alla system, inte vara microsoft, utan vår

lifecycle management process drar ju med sig alla system egentligen och det är något vi måste ha koll på.

25. A: Ja, jag tänkte höra om ni har några särskilda strategier som ni använder gällande hanteringen av teknisk skuld, till exempel om ni prioriterar skulderna på något sätt?

26. L: Jo men man har ju alltid en dialog med verksamheten. Om man har ett system som funkar så kan man ibland känna att varför ska jag som systemägare eller som ansvarig för en del av verksamheten lägga pengar på att uppgradera ett system som redan fungerar och innehåller de möjligheter och funktioner som jag vill ha. Ibland så måste man då ha en lite mera tydlig dialog med verksamheten tala om att för att vi inte ska hamna efter, alltså ålägga oss med teknisk skuld eller vad det kan va för någonting, eller att de blir out of date och inte supporterade längre, så måste du lägga pengar på att uppgradera. Du kanske inte får så mycket funktionalitet eller det kanske inte blir så stor nytta för dig.

27. A: Alltså bara ren underhållning egentligen?

28. L: Ja precis, och det är alltid en tråkig kostnad om man nu är från verksamhetens sida. Där får man alltså ha en dialog hela tiden och att börja i god tid. Det går inte att komma och säga till att om en månad så kommer vi stänga ner din server. Det går ju inte utan ofta har man ju kopplingar till andra system som man måste byta ut också, så det brukar ofta vara en kedja av händelser som även kostar pengar, så en viktig del av egen erfarenhet är att det behöver förankras i god tid för att man ska hinna byta ut systemen. Så strategin är egentligen att ha en god dialog med verksamheten och även att börja i god tid. Och lite personlig reflektion på det här är att när det handlar om säkerhet så är det också väldigt ofta en tråkig utgift och det märks ju ofta inte så tydligt. Men där får man ju försöka hitta "what's in it for you", alltså, om jag gör det här, vad får du ut av det och hur gynnar det dig. Så jag jobbar ganska mycket med att ha den här approachen när jag vill göra förändringar inom säkerhetsområdet, alltså att hitta selling points och USPar i allmänhet där man kan se hur det blir bättre för dig om vi gör så här. Det blir enklare för dig, det går snabbare eller liknande. Dina medarbetare kan känna sig tryggare för att de kan inte göra fel och så vidare. Så det är lite grann hur man lägger fram det, men det är lite mer personlig reflektion.

29. Lo: Men när du tar upp det så är det intressant för att vi har läst lite om avdelningar som kanske inte har en förståelse för teknisk skuld och hållbarhet rent systemmässigt så att säga, och att de på sitt sätt kan leda till en ökad teknisk skuld i och med att de kanske inte riktigt förstår värdet i det. Med andra ord, kan du se att andra avdelningar föder på den tekniska skulden på grund av att de inte har en förståelse för förändringarna så att säga?

30. L: Ja, så kan det vara och det kan ju också vara så att många gånger sitter kanske verksamheten på budgeten och därmed pengarna. Det är ju deras budget som pengarna tas ifrån för att uppgradera ett system, och pengar är ju i ofta så att säga makt. Om jag sitter som kravställare men inte har några pengar eller ingen budget, och så ska jag tala

om för någon att du måste lägga XX pengar på det här. Då lär man inte få någon nytta, man kommer inte märka någon skillnad och du behöver bara göra det. Det är alltså ett svårare läge. Det var därför jag menar att här bör man vara lite Street smart eller man måste ha lite psykologi lite grann och säga typ vad kan jag plocka fram för fördelar för dig, innan jag går in i dialogen och säger "det här ska du bara göra". Det är väldigt dumt sätt att göra det man jobbar med säkert. Däremot kan jag ju alltid dra det kortet, alltså säkerhet är så viktigt. Jag kan gå upp ett steg och säga att om vi inte gör det här så så kommer vi att öka vår risk för ett angrepp eller för en hacker. Och det är ju jobbigt att ta det beslutet och säga att "jag vill inte lägga pengar här, men jag riskerar att sänka eller företaget om vi får en ransomware attack". Det är ju självklart svårt och det är inte så jag tycker det är konstruktivt att jobba på längre sikt, utan det är mer att hitta gemensamma fördelar över varför vi ska göra det här osv. Jag menar, du kan ju också kanske få möjlighet till ny funktionalitet när du uppgraderar. Du kanske inte kan nyttja det från dagen, men gör vi den här förändringen så öppnar vi upp möjligheten för att göra olika saker som på sikt kanske kan hjälpa dig och din avdelning till att göra någonting bättre.

31. Lo: Lite mer allmänt. Finns det någonting du känner att ni behöver ändra på eller som ni borde ändra på gällande er hantering av teknisk skuld?

32. L: Nah, det blir lite mer generellt som vi inne på tidigare att man måste ju börja tidigt med de här frågorna och man måste ha en bra dialog med den som sitter med pengarna. Det är ju det som i min värld är det absolut viktigaste. Alltså, för att det ska bli en bra affär så bör båda känna en slags win-win känsla. Båda ska ju vara nöjda när man går därifrån. Om man nu drar någon analogi eller parallell till om du ska köpa en vara eller sälja någonting, att om både säljare och köpare är nöjda, då blir det ju en väldigt bra affär. I min värld så får man försöka hitta det, hur ska jag hitta så att vi båda blir nöjda. Där jag får som jag vill och jag kan övertyga min motpart om att det här är en bra grej också. Så då är självklart en god ide att börja i god tid, plus att jag måste ha en förståelse för, och lyssna på verksamheten, om jag kommer från det här hållet då. För ofta drivs de här från IT, alltså de här uppgraderingsprojekt, och då är det ju det här klassiska om att nu kommer de här IT med sina jobbiga grejer. Det kommer bli störningar, det kostar och det blir svårare att navigera osv. Så ofta kan det ju finnas ett motstånd för verksamheten, att det kanske har misslyckats förut eller blivit varit dåligt förut just för att det drivs lite från fel håll. Så min personliga approach till det här är ju det jag försöker beskriva att om man ska lyckas och ha en bra långsiktig relation till verksamheten, eller vem det nu är som man vill göra någonting med, så är det bra om man tänker lite win-win.

33. O: Vi kan väl egentligen komma in lite mer på arbetsmetoder och system. Vad har ni för olika system och hur är de sammankopplade? Väldigt grovt beskrivet.

34. L: Ja väldigt grovt beskrivet kan jag säga som jag tidigare beskrivit att vi håller på att göra en resa där vi tar oss från en miljö till en annan. Så att väldigt grovt kan man säga att vi har ett färre antal nyckelsystem som egentligen hanterar det mesta i banken. Det blir så att säga "klumpar" som sen har en massa kopplingar hit och dit, och sen så ställer man frågor till databasen om vad man ska ha. Beskrivet med väldigt grova drag så att säga. Så att dit vi är på väg är att vi håller på att sticka upp det här med mikrotjänster. Vi använder

ju den typen av teknik också, så att vi kommer ju att splitta upp hela våran, så att säga grundstruktur och bygga små byggklossar istället, små mikrotjänster. Så vi kommer ju bygga hela våran nya IT infrastruktur på en ny teknik.

35. A: Det är med andra ord en omstrukturering som ni är mitt uppe i där ni byter arkitektur?

36. L: Precis! Så egentligen är det ett ganska stort jobb som vi håller på med just nu, att byta ut hela den arkitektur som finns och istället bygga små, sammankopplade tjänster runt det.

37. A: Och hur ser du på det bytet av arkitektur till MSA?

38. L: Det ställer ju utmaningar på hela organisationen, i synnerhet för mig och IT och även hela projekt organisationen också. Vi kommer ju använda en ny teknik, så det ställer ju även krav på både mjukvara och teknik och sådana saker, och det kanske då även öppnar upp för nya tjänster inom Microsoft. Vi rör ju oss långsamt mot molnet så att säga och där finns ju många Cloud tjänster som vi också är intresserad av att titta på eftersom Cloudtjänster har många fördelar. Det går ju snabbare att implementera och så vidare. Det finns en större kompetens runt där så att vi rör oss ju också åt det hållet och det ställer ju utmaningar för mig runt säker som utveckling, att vi har de verktyg vi använder när vi bygger och när vi kontrollerar våra tjänster eller byggen att de kan se liksom det nya vi håller på med.

39. A: Då får man använda sig mer av tredjepart-leverantörer också kan tänka mig.

40. Ja det ja, det kanske blir så. Det kan man väl kanske kalla det för. Vi måste hitta pålitliga system, alltså. I det finns ju hur många Cloud tjänster som helst och det poppar väl säkert upp tusentals varje dag. Så här handlar ju om att hittar rätt tjänst för rätt sak vid det här tillfället och och det ska också vara lite långsiktigt. Det kan ju poppa upp en app som känns ascool i idag, men. Om om några veckor så nej, det var inte så bra så att och eftersom vi är en bank så kan ju inte vi liksom vara betatestare på det sättet utan vi behöver pålitliga system som som vi kan känna oss trygga med så att det det handlar om att man rör sig i en kontrollerad fart eller kontrollerar takt utåt och så måste man ha kontroll. Eller om vi liksom vad är det för ny teknik eller ny tjänst vi använder? Hur är den uppbyggd? Vad finns för kontrollmekanismer i det här? Man måste hela tiden titta på the darkside så att säga. De följer ju utvecklingen också har nya verktyg för att hacka sig in så att det är ju viktigt att man har koll på sin säkerhet.

41. O: Jag antar att ni har tänkt på det ganska mycket och gått igenom det. Men nackdelar med användandet av mikrotjänster är i allmänhet för det finns mycket fördelar, men man pratar mycket om fördelar och sällan egentligen och nackdelarna.

42. L: Ja, det är det som jag menar du flyttar ju ut, du flyttar ju ut i molnet, det är det man gör. Det finns ju både för och nackdelar med molnet så att det här handlar om att du

lämnar ifrån dig lite grann av din. Ja, din kompetens och din verksamhet hos någon annan då egentligen. Men som jag sa är fallet så vi är ganska Microsoft-tunga så att vi känner ju att vi rör oss liksom långsamt åt det hållet. Men vi känner ju också att vi har liksom kontroll på det vi gör och när vi gör det och vi jobbar ju mycket med de här säkerhetssystem som Microsoft har också alltså när vi scannar, behavior-analytics såna här saker så att vi måste ju haka på den typen av tjänster också, alltså säkerhetstjänsterna måste också kunna hänga på när vi flyttar ut oss molnet. Så mikrotjänster är ju för det mesta bra så länge du har kontroll på vad du gör, om man nu ska ju se det på det sättet. Det kan ju gå väldigt fort också om du släpper för mycket. Alltså våra utvecklare är väldigt kreativa och som sagt det kommer en massa coola grejer hela tiden som man vill använda och där, återigen gäller att ha en väldigt bra dialog och vara nära våra utvecklare för att hänga med i vad de gör och hela tiden ha en konstruktiv dialog så att vi skulle vilja när de säger att "vi skulle titta på det här" eller göra på det här sättet måste ju vi som jobbar med säkerhet hänga med och se liksom vad innebär det? Vad kan det liksom innebära? För ni har risker, så det är en utmaning när man förflyttar sig, oavsett om vi kallar för mycket mikrotjänster eller någonting annat att vi måste ju hänga med liksom övriga organisationer också när det gäller compliance och den biten också.

43. Lo: När det äldsta system ifrån som ni använder utav?

44. L: Alltså, jag vet faktiskt inte hur gammalt är svårt att säga vad du menar med gammalt, men. Vi har ju system som funnits länge men som hela tiden uppdateras. Men jag menar, när startade Microsoft nu är det kanske taskig, men använder du Microsoft grejer så har du ju använt Microsoft grejer liksom sen de startade sen nittioalet men alltså vi har som jag sa, vi är ganska unga relativt sett en ganska ungt företag så vi har inga gamla system på det sättet som vi har liksom släpat med oss sen stenåldern, utan kanske max 10 år.

45. Lo: Kan du se en koppling till att det underlättar just hanteringen av teknisk skuld eller att det minskar den tekniska skulden?

46. L: Att förnya dina system regelbundet är ju en fördel.

47. O: Eftersom att legacy då kan bidra till svårigheter med hanteringen av tekniskt skuld?

48. L: Jo men så är det helt klart och det vet jag ju. Mina kollegor i branschen som jobbar på storbankerna till exempel. Som har extremt stora projekt som måste dras igång som kostar otroligt mycket pengar för att kunna förändra vissa saker om vi mappar tillbaka med PSD2 då. För att då deras stordatorer i de här storbankerna ska kunna överhuvudtaget hantera det som i det nya regelverket innebär så satsar de otroligt mycket pengar och stora projekt för att kunna överhuvudtaget få till det. Så att det är väldigt kostsamt om man liksom låter det gå för länge så jag tror inte det är någon bra idé utan återigen har man en bra dialog med verksamheten och ser till att de hela tiden följer någon typ utav livscykel management så så bör man inte hamna där. Så där men samtidigt regelverk kan ju slå ner lite grann. Det är inte marknaden som styr utan det är

regelverket som styr så att ibland är det ju så att du som organisation inte har något val utan du måste ju bara. Inom ett år så ska det funka punkt och då är det ju ett annat läge än om du har en strategi och rör dig mot det här marknadssegmentet eller vad det nu vill göra och öppna upp för de här tjänsterna och då blir ju projekten mera, vad ska man kalla det regelstyrda istället för marknadsstyrda? Och det är inte bra heller för då kanske det inte gör det bästa för organisationen, men men du gör det för att kunna liksom bli compliant. Nej, det är aldrig bra att ha för mycket legacy eller så mycket teknisk skuld. Det kan jag inte påstå är bra någonstans.

49. O: Nej det, det är väl det vi också har kommit fram till.

50. Lo: Om vi rör oss vidare lite bara så här: Arbetar ni med med agila arbetsmetoder idag i de teamen du ansvarar för så att säga.

51. L: Yes, det gör vi. Vi jobbar ju ganska mycket med safe-ramverket om ni har tittat någonting på det så vi jobbar ju mycket med det devops och mycket med agila arbetssätt och så vidare med de här. Och jag tycker att det är ju funkar bra, bara man fortfarande har koll på vad du gör. Det är ju samma sak i våra sprintar. Men när vi jobbar med safe-ramverket så blandar vi in verksamheten på ett helt annat sätt än man gjorde tidigare. De får vara med hela tiden genom demos och alltihopa och får se och får komma med in på hela utvecklingen. I de här små sprintarna som vi gör då. Så att det gör ju att verksamheten är med på ett helt annat sätt så att, utifrån ett safe ramverk så tycker jag att det funkar bra med de agila metoderna. Tittar vi tillbaka ett antal år när det kom så var det mera, typ några lata kodare som hade kommit fram till agilt för det var jobbigt att dokumentera, “ jag får göra som jag vill så blir det bra ungefär”. Nu raljerar jag jättemycket, men om ni förstår hur jag menar. Men tar du godheterna i den agila arbetssättet och så kopplar du på det till safe och devops också tror jag att det är bra. Jag tycker att vi har hittat en bra modell för det hos oss.

52. O: Så med andra ord det underlättar hanteringen av egentligen en vidare komplexitet i systemen och i allmänhet.

53. L: Alltså jag jämför, jag har ju varit med ganska länge så jag jämför de här traditionella projekt, de här projekten som man körde förut då med med PN 3 eller vad man nu hade för modell. Där du får en uppgift av en beställare och sen så levererar det ett halvår senare och beställaren trodde att han skulle få en blå boll, men han fick en röd fyrkant för att det blev så och han fick inte vara mer längs vägen så att jag tycker det är en bättre metod som vi använder nu för tiden, där vi ska ta fram, göra förändringar i våra tjänster eller göra förändringar över huvud taget. Och det går ju också mycket fortare idag så att det behöver ju ändras. Förr i tiden som den där om du använder gamla projekt metoden idag så är det ju så när du väl har lagt din beställning så egentligen redan då så vet du att det är att inte det du vill ha för att. Det kommer springa ikapp dig längs vägen så om ett halvår så stämmer inte beställningen med vad du faktiskt behöver. Så att ha möjlighet att

hela tiden svänga åt kika lite på i det korta perspektivet tror jag är nödvändigt för att möta liksom den moderna digitaliseringen.

- 54. O: Ja, jag tror faktiskt att det var de frågorna som vi tänkte ställa. Vi brukar avsluta egentligen med så här nu. Nu är inte Carnegie direkt utsatt för detta, men hur tror du konkurrensen då mellan bank aktörerna i allmänhet. Tror du konkurrensförutsättningarna ändrats i och med PSD2?**
55. L: Alltså jag. Min personliga spaning är ju att jag trodde att det skulle hända mer. Jag tror det skulle komma lite flera såna här aktörer med lite mera coola bank appar om man kallar det för det när du nu har möjligheten att koppla ihop en persons olika engagemang hos olika bankinstitut och koppla ihop det till en app som man kan få ett coolt gränssnitt med bra information. Jag jag tycker inte jag sett någonting sånt än som gör att jag blivit imponerad så som jag trodde jag skulle bli. Det har gått ganska segt, jag vet inte varför, men jag förväntar mig att det skulle gå lite snabbare om man säger så sen får vi se vad som ligger liksom runt hörnet, men hittills och vet inte.
- 56. O: Det är väl alltid lite så att Gartners hype-cycle när det gäller såna här förändringar också? Man tror att det kommer göras stora förändringar och sen så går det lite långsammare än man trodde.**
57. L: Ja precis, och jag vet inte heller om det beror på att man. Alltså, mina pengar är ju rätt viktigt för mig att ha koll på dem. Jag måste nog lita på den som har koll på min plånbok så att säga och då känns det kanske fortfarande såhär att de här storbankerna de är nog ganska bra. "Ska jag lägga mina pengar hos Google eller någon annan sådan här nisch-aktör, njaa.. vet inte. Jag vill nog kanske ha dem hos en storbank än så länge i alla fall".
- 58. O: Tusen tack Lennart. Väldigt lärorikt och intressant att lyssna på dig och vi uppskattar det otroligt mycket. Så tack.**
- 59. Lo: Ja verkligen.**
60. L: Absolut, och lycka till nu med uppsatsen.
- 61. O: Ja, men absolut vi. Vi hörs av helt enkelt när det. Vi skickar gärna över innan så du får läsa uppsatsen.**
62. L: Ja, det vore kul att se. Jag tar gärna del av det ni kommer fram till det vore det är spännande så att lycka till.
- 63. O: Men absolut tack så mycket. Ha det gott.**

8 Referenser

- Allman, E (2012). *Managing Technical Debt*.
Tillgänglig: <https://dl.acm.org/doi/10.1145/2160718.2160733> [Hämtad: 26/3-21].
- Alves, N. S. R., Mendes, T. S., de Mendonça, M. G., Spínola, R. O., Shull, F. & Seaman, C. (2016). *Identification and Management of Technical Debt: A Systematic Mapping Study*, *Information and Software Technology*. Tillgänglig: <https://www.sciencedirect.com/science/article/abs/pii/S0950584915001743> [Hämtad: 27/3-21]
- Avanza (2021). *We started Avanza with a simple idea*
Tillgänglig: <https://investors.avanza.se/en/about/about-avanza/> [Hämtad 22/3-21]
- Bennet K, (1994). *Legacy Systems: Coping with Success*.
Tillgänglig: <https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=363157> [Hämtad 1/4-2021]
- Bryman A, Bell E (2012). *Social Research Methods*. (4:e uppl.). Oxford: Oxford University Press
- Codabox Z & Williams B (2013). *Managing Technical Debt: An Industrial Case Study*.
Tillgänglig: https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=6608672&casa_token=uNLgIM0kQnEAAAAA:kBkFpNJ7cxxkz0x2qXrFX16Jfe7vf86c8DI1DtOxFjly4r3FJ-oGZB0rTl_KktQOxKNdxClTow&tag=1 [Hämtad 19/4]
- Deloitte (2020). *Opening banking through architecture re-engineering A microservices-based roadmap*
Tillgänglig: <https://www2.deloitte.com/content/dam/Deloitte/us/Documents/financial-services/us-enabling-platform-banking-pov.pdf> - sida: 2 [Hämtad: 21/4-21]
- Edith T, Aybuke A, Vidgen R (2013). *An exploration of technical debt*. Publicerad: Elsevier.
Tillgänglig: <https://www.sciencedirect.com/science/article/abs/pii/S0164121213000022> [Hämtad: 26/3-21].
- Eriksson K, (2019). *Neobankerna är här*
Tillgänglig: <https://www.kth.se/aktuellt/nyheter/neobankerna-ar-har-1.923592> [Hämtad 15/3-21]
- Fowler M, (2019). *Technical Debt*.
Tillgänglig: <https://www.martinfowler.com/bliki/TechnicalDebt.html> [Hämtad 14/4-21]
- Finansinspektionen, (2019). *Betaltjänstdirektivet PSD2*.
Tillgänglig: <https://www.fi.se/sv/bank/andra-betaltjanstdirektivet-psd-2/> [Hämtad 22/3-21].
- Jacobsen, D. I., & Sandin, G. (2002). *Vad, hur och varför: om metodval i företagsekonomi och andra samhällsvetenskapliga ämnen*. Lund: Studentlitteratur.
- Konkurrensverket, (2018). *Konkurrensen i Sverige 2018, kap 9. Bankmarknaden*
Tillgänglig: https://www.konkurrensverket.se/globalassets/publikationer/rapporter/rapport_2018-1_kap9-bankmarknaden.pdf [Hämtad 11/3-21].
- Kreditkoll (2021). *Neobanker*
Tillgänglig: <https://www.kreditkoll.nu/neobanker> [Hämtad: 15/3-21]
- Lehman MM, Ramin J F & Wernick P D (1997). *Metrics and Laws of Software Evolution - The*

Nineties View.

Tillgänglig:

https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=637156&casa_token=qyn20IJUf1wAAAAA:0MLv-DMiF-hq7hADmu1VS7NCh6D65Zv6k-UxZC_ngFQtdrXeBVBHbIIYUNhhurNQFjipJ-1f5Tg&tag=1 [Hämtad 22/3-21]

Li, Z., Avgeriou, P. & Liang, P. (2015). *A Systematic Mapping Study on Technical Debt and Its Management*, The Journal of Systems & Software.

Tillgänglig: https://www.researchgate.net/publication/269396520_A_Systematic_Mapping_Study_on_Technical_Debt_and_Its_Management [Hämtad: 27/3-21]

Nationalencyklopedin (2021). *Strukturomvandling*

Tillgänglig:

<https://www.ne.se/uppslagsverk/encyklopedi/l%C3%A5ng/strukturomvandling> [Hämtad: 25/4-21]

Nicoletti Bernardo (2017). *The Future of Fintech - Integrating finance technology in financial services*

Tillgänglig: <https://link.springer.com/content/pdf/10.1007/978-3-319-51415-4.pdf> [Hämtad: 28/4-21]

Open Banking (2021). *What is Open Banking?*

Tillgänglig: <https://www.openbanking.org.uk/customers/what-is-open-banking/> [Hämtad 21/4-21]

Premchand Anshu, Anurag Choudhry, 2018. *Open Banking & APIs for Transformation in Banking*. Publicerad: IEEE

Tillgänglig: <https://ieeexplore.ieee.org/abstract/document/8668107/authors#authors> [Hämtad 22/3-21].

Raune. M Janet, (2006). *A och O i samhällsvetenskaplig forskning*. Publicerad: Studentlitteratur, Lund.

Riksdagen (2008). *Permanent förändring - Globalisering, strukturomvandling och sysselsättningsdynamik* (Bilaga 7 till Långtidsutredning)

Tillgänglig: <http://data.riksdagen.se/dokument/GWB321> [Hämtad: 22/4-21]

Sandström C, Konkurrensverket (2017). *Disruptiv digitalisering inom marknaden för juridiska tjänster* Tillgänglig:

https://www.konkurrensverket.se/globalassets/publikationer/uppdraagsforskning/forsk_rapport_2017-4.pdf [Hämtad 24/3-21]

Sharma Tushar, (2018). *Four strategies for managing technical debt*. Tillgänglig:

<http://www.designsmells.com/articles/four-strategies-for-managing-technical-debt/?fbclid=IwAR1VR5KqTUxM-9ZkJOv00oy-AfzwUq5dca2Mk9yOdGBTJkfCZ9-C40sncj0> [Hämtad 27/3-21]

Suryanarayana et al. (2015). *Refactoring for Software Design Smells: Managing Technical Debt* Publicerad: Elsevier. Tillgänglig:

https://books.google.se/books?hl=sv&lr=&id=1SaOAwAAQBAJ&oi=fnd&pg=PP1&dq=Suryanarayana+et+al.+2015+technical+debt&ots=OxcXEH4sHc&sig=FDTim3yc4j1VR_A_bK3dH7OudgCc&redir_esc=y#v=onepage&q=Suryanarayana%20et%20al.%202015%20technical%20debt&f=false [Hämtad 27/3-21].

Svenska bankföreningen 1, (2020). *Bankerna i Sverige*.

Tillgänglig: <https://www.swedishbankers.se/fakta-och-rapporter/svensk-bankmarknad/bankerna-i-sverige/>

- [Hämtad 11/3-21].
Svenska bankföreningen 2, (2020). *Bankstrukturen*.
Tillgänglig: <https://www.swedishbankers.se/fakta-och-rapporter/svensk-bankmarknad/bankstrukturen/> [Hämtad 11/3-21].
- Svenska Bankföreningen (2016). *Flera initiativ kring digitalisering*
Tillgänglig: <https://www.swedishbankers.se/repository/bankfokus/bankfokus-nr-4/flera-initiativ-kring-digitalisering/> [Hämtad 22/3-21]
- Svenska Bankföreningen (2019). *Vad digitaliseringen gör med konkurrensen på finansmarknaden* Tillgänglig:
<https://www.swedishbankers.se/repository/bankfokus/bankfokus-nr-3-2019/vad-digitaliseringen-goer-med-konkurrensen-paa-finansmarknaden/> [Hämtad 22/3-21]
- Tobias Modig (2018). *Uppdragsrapport: Ut med Cobol, in med microservice ... och släng det aktivitetsbaserade kontoret i sopkorgen!*
Tillgänglig: <https://www.citerus.se/ut-med-cobol-och-aktivitetsbaserade-kontor/>
[Hämtad: 21/4-21]
- Wallström, (2005). *Stort missnöje med banksystem*.
Tillgänglig: <https://computersweden.idg.se/2.2683/1.13897/stort-missnoje-med-banksystem?fbclid=IwAR3nFjusad8smUbqop6PbmTVebE-k6LeidYuUMSMFJD9AvnmJDN3wEgcUFc> [Hämtad 30/3 - 21]
- Warren I, (1999). *The Renaissance of Legacy Systems - Method Support for Software System Evolution*. Tillgänglig:
https://books.google.se/books?hl=sv&lr=&id=KevSBwAAQBAJ&oi=fnd&pg=PA1&dq=replacing+legacy+systems&ots=zOfK1w6gMW&sig=rUQ2hwdP_JOXIJwRtuyhgfp5LHw&redir_esc=y#v=onepage&q&f=true [Hämtad 18/3 - 2021]
- Yves L. Doz & Keeley Wilson, (2018). *Ringtone: Exploring the Rise and Fall of Nokia in Mobile Phones*. Tillgänglig: <https://journals.sagepub.com/doi/abs/10.1177/0001839218791184>
[Hämtad 20/4-21]