Linköpings universitet | Institutionen för datavetenskap
Examensarbete på avancerad nivå, 30 hp| Kognitionsvetenskap
Vårterminen 2019 | LIU-IDA/KOGVET-A--19/011--SE

Hur kan en chatbot implementeras som ett hjälpmedel för supporttekniker?

- En förstudie för att undersöka möjligheten att implementera en chatbot i en industriell verksamhet

Ellinor Ihs Håkansson

Handledare: Annika Silvervarg
Examinator: Arne Jönsson

Extern handledare: Erik Carlsson

LINKÖPINGS UNIVERSITET

Upphovsrätt

Detta dokument hålls tillgängligt på Internet – eller dess framtida ersättare – under 25 år från publiceringsdatum under förutsättning att inga extraordinära omständigheter uppstår.

Tillgång till dokumentet innebär tillstånd för var och en att läsa, ladda ner, skriva ut enstaka kopior för enskilt bruk och att använda det oförändrat för ickekommersiell forskning och för undervisning. Överföring av upphovsrätten vid en senare tidpunkt kan inte upphäva detta tillstånd. All annan användning av dokumentet kräver upphovsmannens medgivande. För att garantera äktheten, säkerheten och tillgängligheten finns lösningar av teknisk och administrativ art.

Upphovsmannens ideella rätt innefattar rätt att bli nämnd som upphovsman i den omfattning som god sed kräver vid användning av dokumentet på ovan beskrivna sätt samt skydd mot att dokumentet ändras eller presenteras i sådan form eller i sådant sammanhang som är kränkande för upphovsmannens litterära eller konstnärliga anseende eller egenart.

För ytterligare information om Linköping University Electronic Press se förlagets hemsida http://www.ep.liu.se/.

© Ellinor Ihs Håkansson

Sammanfattning

Syftet med det projekt som beskriv i denna uppsats är att undersökta möjligheterna med att implementera en chatbot i Siemens verksamhet. Fokuset för chatboten skulle vara att den på något sätt skulle implementeras som ett hjälpmedel för supportteknikerna på Siemens. Supportteknikerna arbetar dygnet runt med kundärenden som kommer in rörande deras maskiner. Därmed behöver supportteknikerna ibland hitta dokumentation och information kring dessa maskiner för att kunna ge rekommendationer till kund om hur det eventuella problemet ska lösas.

I denna studie gjordes en litteraturundersökning kring chatbotar i allmänhet, tekniken kring chatbotar idag och hur en chatbot bör designas. Dessutom genomfördes en datainsamling i form av kontextuella intervjuer som sedan strukturerades till ett affinitetsdiagram. Detta affinitetsdiagram var i sin tur grunden för den persona och det kontextscenario som skapades. Affinitetsdiagrammet och de kontextuella intervjuerna beskrivs i mer detalj i arbetet skrivet av Linn Olsson (2019). Utifrån personan, kontextscenariot, affinitetsdiagrammet samt bakgrundslitteraturen skapades en uppsättning krav. Dessa var krav kring hur chatboten skulle se ut och fungera. Det som eftersöktes mest var något som kopplade samman alla system de använde sig av för att hitta information kring de olika maskinerna, samt en enklare sökfunktion för att hitta specifik dokumentation kring ett visst taggnamn.

Kraven blev sedan poängsatta och prioriterade av supporttekniker på Siemens. De prioriterade kraven var utgångspunkten i designandet av en prototyp av en chatbot. Detta implementerades sedan i en interaktiv prototyp. Den prototyp som skapades är ett första utkast på hur en chatbot skulle kunna se ut för just uppgiften att hitta specifik information. Denna uppgift var som tidigare nämnts något de uttryckte sig vara i störst behov av. Detta förslag på användningsområde för en chatbot i Siemens system hade troligtvis underlättat supportteknikernas arbete i ganska stor grad. Det minskar tiden de lägger på att försöka hitta informationen så att de snabbare kan lösa kundärenden. Det här visar på att det finns möjligheter att implementera en chatbot i deras system. Dock krävs ytterligare undersökningar kring exakt vad för information supportteknikerna vill åt, samt hur det ska möjliggöras att de kan få tag på denna information, för att kunna implementera en chatbot. Dessutom skulle de behöva undersöka fler uppgifter som denna chatbot skulle vara användningsbar för. Detta för att chatboten får ett större användningsområde och kan på så vis få en större avgränsning från att likna en begränsad sökmotor. På så vis skulle detta vara ytterligare argument för att en chatbot bör implementeras.

Förord

Jag vill först och främst tacka Erik Carlsson och Per Södergren för möjligheten att få göra detta projekt med och för er, samt för allt stöd och all hjälp Erik har gett under projektets gång. Därefter vill jag tacka Annika Silvervarg för att hon varit en sådan bra handledare som hjälp mig, väglett mig och stöttat mig genom hela processen. Sedan vill jag tacka Linn Olsson som jag inledde detta projekt med, vars uppsats jag refererar till ett antal gånger i denna uppsats. Jag vill tacka henne för det stöd och den peppning jag fått samt att jag fått göra stora delar av arbetet tillsammans med henne. Jag vill även tacka Mattias Arvola för all hjälp jag fått, speciellt kring hur jag skulle bygga upp min uppsats, samt min kravformulering.

Slutligen vill jag ge ett stort tack till alla i masterrummet(-en) för allt stöd, all hjälp jag fått och den härliga tiden vi haft tillsammans. Utan er hade det varit mycket tuffare att ta sig igenom detta. Ingen nämnd ingen glömd, ni vet vilka ni är!

Ett stort tack till alla, det hade inte varit möjligt utan er!

Linköping i maj 2019

Ellinor Ihs Håkansson

Innehållsförteckning

1. Introduktion 9

1.1 Syfte 10

1.2 Frågeställning 10

2. Teori 11

2.1 Tidiga och regelstyrda chatbotar 11

2.2 Chatbotar baserade på maskininlärning 15

2.3 Chatbotar och dess funktionalitet idag 17

2.4 Chatbotens uppbyggnad och design 21

2.4.1 Designriktlinjer 22

3. Metod 29

3.1 Kontextuella intervjuer och affinitetsdiagram 29

3.2 Kravformulering 30

3.2.1 Persona 31

3.2.2 Kontextscenario 32

3.2.3 Kategorisering av krav 33

3.2.4 Prioritera krav 34

3.3 Skissning av prototyp 35

3.4 Prototypningsverktyg 36

4. Resultat 37

4.1 Affinitetsdiagram 37

4.2 Kravformulering och prioritering 42

4.2.1 Persona och kontextscenario 42

4.2.2 Sammanställda och prioriterade krav 44

4.3 Slutgiltig prototyp 49

5. Diskussion 55

5.1 Affinitetsdiagram, persona och krav 55

5.2 Prototyp 58

5.2.1 Prototyp i förhållande till prioriterade krav 58

5.2.2 Prototypens möjligheter 59

5.3 Framtida funktion och dagens teknik 60

5.4 Möjlig framtida implementering 62

6. Slutsats 65

Referenser 67

Appendix A – Fullständig prototyp, steg för steg 71

1. Introduktion

Chatbotar är idag en teknik som blivit väldigt populär att implementera, där de exempelvis används som ett supportverktyg, i utbildningssyfte eller till och med enbart som något underhållande (Io & Lee, 2018). De har ett brett användningsområde och underlättar arbete genom att kunna ta över repetitiva uppgifter. Istället för att exempelvis en person som arbetar inom support ska besvara samma typ av fråga flera gånger om dagen så skulle en chatbot kunna hantera dessa frågor och ge svar dygnet runt. En chatbot blir inte heller begränsad till att enbart kunna svara en person åt gången, utan kan användas av flera personer samtidigt. En chatbot blir på så vis kostnads- och tidseffektivt samtidigt som den minskar arbetsbördan för den person som i detta exempel arbetar med support.

De chatbotar som finns idag är bland annat Apple Siri, Telias supportchatbot, H&Ms kik bot och många fler. Alla dessa har olika syften men där grundsyftet är att underlätta användarens vardag. De uppgifter som chatbotar är mest lämpade för är enklare uppgifter i form av exempelvis enkla fråga-svar-konversationer eller att leta upp någon typ av information (Zamora, 2017). Idag används oftast maskininlärning då detta bland annat gör det möjligt för dem att hantera och besvara samma typ av frågeställningar även om de är skrivna på olika sätt (Biswas, 2018). Med denna teknik och övrig Artificiell Intelligens-teknik (AI) som finns idag, och som fortsätter att utvecklas, så finns det stora möjligheter både nu och i framtiden vad gäller chatbotar och dess utveckling.

Ett företag som har fått upp intresset för denna teknik och som detta projekt sker i samarbete med är Siemens Industrial Turbomachinery (SIP) AB. Siemens är ett globalt företag och är en av världens största företag när det kommer till hälsovård, energi och industri (Siemens AB, n.d.). Siemens AB består av flera olika bolag, däribland SIP AB. Detta bolag tillverkar ång- och gasturbiner som används över hela världen där de bland annat kan utgöra grundpelaren för kraftanläggningar som ska producera elektricitet, värme eller ånga. De kan även användas som drivkällor för kompressorer och pumpar, främst inom gas- och oljeindustrin. De som köper dessa turbiner får även ta del av en tjänst där de kan få hjälp och support dygnet runt av supporttekniker på SIP AB. Dessa supporttekniker arbetar dagligen med bland annat eventuella problem som uppstår, vid installering av nya turbiner och vid serviceunderhåll av turbiner.

1.1 Syfte

Detta projekt är en förstudie till en studie som Siemens genomför internt. Den studie som genomförs i denna uppsats syftar till att undersöka möjligheten att implementera en chatbot i Siemens system. Detta för att underlätta och effektivisera arbetet för personalen som arbetar på deras support. Denna studie genomförs i samarbete med Linn Olsson (2019) som i sitt projekt utforskar, med hjälp av kontextuell design, hur supportteknikerna arbetar och vad för eventuell problematik som finns med de system de har idag. Denna datainsamling kommer att användas i denna studie tillsammans med information som samlats in via litteratur som handlar om chatbotar, vad för möjligheter som finns med dessa och hur tekniken kring dessa ser ut.

Med hjälp av denna information ska en prototyp framställas som är en mock-ups på hur en chatbot skulle kunna se ut för att både underlätta supportteknikernas arbete och implementera den teknik som finns för chatbotar idag. Diskussion kommer sedan att ske kring hur lämplig denna är att implementera och hur en fullskalig implementering av prototypen möjliggörs. Denna diskussion kommer att utgå ifrån de tekniska system som supportteknikerna använder idag och den chatbotteknik som finns idag. En diskussion kommer även att göras kring möjligheterna i att implementera en chatbot i den undersökta verksamheten.

1.2 Frågeställning

Utifrån projektets syfte kommer dessa frågeställningar att besvaras i denna uppsats:

1. Vilka är förutsättningarna för att implementera en chatbot i Siemens supportverksamhet?
    1a. Vilka krav (data krav, funktionella krav, kvalitativa krav och begränsningar) ställs på en sådan chatbot?
    1b. Hur kan en chatbot utformas så att den möter dessa krav?
    1c. Vilka förutsättningar (organisatoriska/tekniska/designmässiga) krävs eller behövs för att kunna implementera en chatbot?
2. Vilka metoder bör och kan användas för att implementera en chatbot i sin verksamhet?

2. Teori

De senaste åren har chatbotar blivit ett allt mer populärt verktyg som många olika företag använder sig av och implementerar i sina system. Chatbotar används idag för bland annat kundsupport, som hjälp på hemsidor eller exempelvis inom utbildning, där de i vissa fall ersätter mänsklig arbetskraft (Io & Lee, 2018; Shawar & Atwell, 2007). Exempel på stora teknikbolag som äger kända chatbotar som finns idag är bland annat IBM Watson, Apple och Facebook Messenger. Chatbotar finns inom många olika domäner, men de är tyvärr inte väletablerade inom alla. Idag finns det en del chatbotar med fokus på kundtjänst eller snarare FAQ (*frequently asked* questions) som det dessutom finns en del dokumentation och forskning kring. Dessa chatbotar riktar sig i de flesta fall mot kund, där chatbotarna hanterar de frågor som frekvent ställs. Dock finns det tyvärr inte, eller åtminstone väldigt begränsad, dokumentation och forskning inom området teknisk support och mer specifikt i industriella sammanhang, där chatboten är tänkt att användas inom företaget. Därav kommer litteratur mest utgå ifrån allmän kunskap kring chatbotar och till viss del även kring kundtjänstchatbotar som riktar sig mot kunder som är det område som ligger närmast.

I denna del kommer en grundläggande genomgång av chatbotar och chatbot teknik att beskrivas. Detta inleds med att beskriva chatbotens historik och hur regelstyrda chatbotar fungerar för att sedan gå in mer på chatbotar som använder sig av maskininlärning. Dessutom ges det en beskrivning på hur chatbotar fungerar och bör fungera, samt designas, idag.

2.1 Tidiga och regelstyrda chatbotar

Människa-maskin-konversation är en teknologi som använder sig av naturligt språk och olika beräkningsmetodologier för att göra det möjligt och enkelt att konversera och interagera med användare. Chatbotar är en relaterad term till detta och har utvecklats sedan långt tillbaka i tiden (Shawar & Atwell, 2007). Den första skapades nämligen redan år 1966 av Joseph Weizenbaum som han gav namnet ELIZA (Weizenbaum, 1966). ELIZA beskrevs dock inte som en chatbot av Weizenbaum, men som ett program som en användare kunde ha en konversation med via naturligt språk. Hon skulle fungera som en simulation av en psykoterapeut, och utifrån den funktionalitet hon hade skulle hon kunna beskrivas som en chatbot eller dialogsystem (Io & Lee, 2018; Weizenbaum, 1966). Efter ELIZA utvecklades fler chatbotar, eller dialogsystem, som exempelvis PARRY (Colby, 1999), CONVERSE (Batacharia, Levy, Catizone, Krotov, & Wilks, 2013) och ALICE (Abu Shawar & Atwell, 2003).

ALICE grundades av Wallace år 1995 och är en förkortning för *Artificial Linguistic Internet Computer Entity* (Abu Shawar & Atwell, 2003; Shawar & Atwell, 2015). Denna chatbot använder sig av mönstermatchningsalgoritmer och är därmed regelstyrd, vilket innebär att den matchar inputmönster med outputmallar (Shawar & Atwell, 2015). ALICE sparar sina kunskaper om engelska konversationsmönster i något som kallas för AIML-filer (*eng. Artificial Intelligence Mark-up Language*). Nedan ges en beskrivning av vad AIML-filer är (Abu Shawar & Atwell, 2003, egen översatt).

> AIML består av dataobjekt som kallas för AIML-objekt som består av enheter som kallas ämnen (*eng. topics*) och kategorier. Ämnet är ett valfritt toppnivåelement som har ett namnattribut och en uppsättning av kategorier relaterade till det ämnet. Kategorier är den grundläggande enheten av kunskap i AIML. Varje kategori är en regel för att matcha en input och konvertera det till ett output, och som består av ett mönster som representerar den input som användaren gett samt en mall som implicerar ALICE-robotsvaret. AIML-mönstret är simpelt och består enbart av ord, mellanslag samt wildcardsymbolerna _ och *. Orden kan bestå av bokstäver och siffror, men inga andra karaktärer. Ord separeras av ett enda mellanslag och wildcardsymbolerna fungerar som ord. Mönsterspråket är oföränderligt mellan fall.

Det interpreterande programmet för AIML-filerna försöker att hitta den bästa mönstermatchningen vilket normalt sätt brukar vara den längsta och för att göra detta används en djupet-först-sökningsteknik (Abu Shawar & Atwell, 2003). Djupet-först-sökning är en sökstrategi vid användning av så kallade sök-träd som i sin tur är en typ av sökalgoritm. Djupet-först innebär alltså att sökningsalgoritmen alltid expanderar till den djupaste noden för varje gren i sökträdet. När sökningen kommit till en viss grens djupaste nod, men ännu inte funnit en lösning så backar den bakåt till närmsta förgrening med oupptäckta noder och söker sig ner i nästa nod (Russel & Norvig, 2010). Denna process stannar alltså när en matchning har hittats, och därefter bearbetas mallen som tillhör den kategorin för att kunna skapa ett output. För att försöka ge en tydligare bild kommer ett exempel att visas som Abu Shawar och Atwell ger i sin artikel, och hänvisar därför till denna artikel för ökad förståelse kring ytterligare detaljer i funktionaliteten kring detta (Abu Shawar & Atwell, 2003). Exempelvis skulle dessa kategorier finnas[1]:

[1] Citeras från Abu Shawar & Atwell 2003

(1) <category>
<pattern>_ WHAT IS 2 and 2</pattern>
<template><sr/> <srai>WHAT IS 2 AND 2</srai>
</template>
</category>

(2) <category>
<pattern>_ WHAT IS 2 *</pattern>
<template><random>
Two.
Four.
Six.
12.
</random>
</template>
</category>

(3) <category>
<pattern>HALO</pattern>
<template><srai>HELLO</srai></template>
</category>

(4) <category>
<pattern>HELLO</pattern>
<template><random>
Well hello there!
Hi
there.
Hi there. I just wanting to talk.
Hello
there !
</random> </template>
</category>

Sedan skickar en användare in denna fråga: ”halo what is 2 and 2”, vilket leder till att processen kommer vara som följande[2]:

1- Denna input kommer att matcha kategori (1) som delar upp den i två meningar: Mening ett: representeras av <sr/>-taggen som matchar (_) vilket är ordet HALO.

2- Ett atomiskt mönster hittas för HALO och ersätts av HELLO i kategori (3) och matchar det sedan igen med kategori (4); svaret kommer att väljas slumpmässigt från mall-listan av kategori (4). Här hanterar den synonymer eftersom halo och hello har samma output.

[2] Citeras från Abu Shawar & Atwell 2003. Egen översättning.

3- Det interpreterande programmet försöker matcha WHAT IS 2 and 2, inget atomiskt mönster hittas, därför appliceras en *backtracking*-teknik, skannar för att finna WHAT IS 2 AND *; igen ingen matchning, en annan *backtracking* försöker finna WHAT IS 2 *, en matchning hittas i kategori (2) och en respons väljs slumpmässigt från listan.

4- Det interpreterande programmet för AIML kommer att kombinera dessa svar tillsammans och visa upp dem.

Det slutgiltiga svaret som visas skulle därmed kunna bli ”Hi there! Four.”. Dock på grund av randomiseringen som fanns skulle svaret även kunna bli en kombination av de övriga listobjekten som fanns.

ALICE har vunnit flertalet gånger i Loebner Prize tävlingen som är en tävling för att utvärdera maskin-konversations-chatbotar (BBC NEWS, 2004; Shawar & Atwell, 2007). De chatbotar som deltar i denna tävling bedöms av domare som rankar dem efter hur ”naturlig” konversationen anses vara. Denna tävling är inspirerad av Alan Turings Turing test och utvärderar maskinens förmåga att få människor att tro att de pratar med en människa (Shawar & Atwell, 2007). Turing testet är ett tankeexperiment som grundar sig i något som Alan Turing kallade för imitationsspelet (*eng. imitation game*). Imitationsspelet består av tre personer där ena personen agerar förhörsledare och de två andra agerar deltagare där ena är en kvinna och den andra är en man. De två deltagarna befinner sig i separata rum från förhörsledaren. Förhörsledaren får enbart kommunicera med och ställa frågor till de andra två deltagarna med någon form av teknologi för att ta lyckas reda ut vilken av dem som är man och vilken som är kvinna. I Turing testet byts i princip en av deltagarna ut mot en maskin, där förhörsledarens uppgift istället blir att reda ut vilken som är maskin och vilken som är människa. Om maskinen lyckas lura förhörsledaren att den är en människa påstår då Alan Turing att det skulle visa på att maskinen är intelligent (Bermúdez, 2014).

Loebner Prize tävlingen anordnades för första gången av Hugh Loebner år 1991 (AISB, n.d.). Tävlingen innefattar, som i Turing testet, att domarna konverserar med flera olika agenter, som antingen är ett datorprogram eller en människa, via ett tangentbord. De ska i sin tur bedöma i hur stor grad de tror att det eller den de konverserar med är en människa. Domarna rangordnar därefter agenterna från att antas vara minst mänskliga till mest mänskliga, och det datorprogram med högsta median i rankning vinner (Shawar & Atwell, 2007). Tävlingen finns fortfarande idag och hålls varje år (AISB, n.d.).

2.2 Chatbotar baserade på maskininlärning

Förr var konversationer mellan chatbotar och människor inte fullt utvecklade och det användes ofta regelbaserade uttrycksmatchningar. Det innebär, som tidigare nämnts, att chatboten via heuristiska sökvägar väljer ett fördefinierat svar (Biswas, 2018; Io & Lee, 2018). Detta begränsar frågeställningar som kan skrivas in till chatboten. Idag används dock oftast maskininlärning, vilket innebär att de lär sig själva hur de ska tolka input och vad de bör svara utefter det. Detta leder på så vis till att användare istället exempelvis kan skriva en och samma fråga på flera olika sätt men ändå få samma svar (Biswas, 2018).

Vad innebär då att en dator lär sig att tolka input eller specifika uppgifter? Det går att jämföra med exempelvis att ett barn ska lära sig att koppla ihop det fysiska objektet som är en stol med själva konceptet av en stol. Att bara säga till barnet att objektet är en stol räcker inte, barnet måste hitta ett sätt att extrahera relevanta egenskaper hos objektet för att kunna karaktärisera det som en stol. För att bygga upp en klassificeringsfunktion förlitar sig människan på egenskaper, även kallat attribut eller variabler. För att en dator i sin tur ska lära sig en viss uppgift så måste det först och främst finnas en uppsättning träningsdata. Denna träningsdata ska innehålla en uppsättning av exempel för det den ska lära sig. Till exempel om datorn ska lära sig alfabetet så innehåller exemplen egenskaper som ska representera en given bokstav. Varje exempel har en tillhörande outputvariabel, alltså i detta fall bokstaven som egenskaperna ska representera. Med hjälp av en klassificeringsalgoritm kan träningsdatat bearbetas vilket leder till att den bästa klassificeraren bör hittas för det specifika problemet. Detta kallas även för övervakad inlärning. Det finns dock något som kallas för oövervakad inlärning men detta kommer inte att beskrivas i denna uppsats (Fernandes de Mello, Antonelli Ponti, Fernandes de Mello, & Antonelli Ponti, 2018).

Maskininlärning är idag vanligt inom *Natural Language Processing*, NLP (Biswas, 2018). Vid utveckling av chatbotar är det oftast NLP som används (Io & Lee, 2018). NLP är både ett forskningsområde och en applikation som både utforskar hur, samt används för att, datorer ska kunna förstå och manipulera naturligt språk och tal. För att göra detta möjligt har och fortsätter forskare att utforska hur människan förstår och använder språk för att sedan applicera detta på datorer. Utifrån detta har man kunnat extrahera tre stora problem som måste hanteras för att kunna bygga datorprogram som förstår sig på naturligt språk. De har att göra med tankeprocessen, representationen samt betydelsen av det lingvistiska input som ges, och till sist

världskunskap (Chowdhury, 2005). NLP tillåter alltså chatboten att tolka vad du skriver och/eller säger till den och därmed ger ett lämpligt svar tillbaka (Phillips, 2018).

För att visualisera hur formaten på träningsdata kan se ut kommer exempel att visas på hur det ser ut för Rasa NLU (*Natural Language Understanding*). Det format som träningsdatat har är en lista med yttranden kommenterade med avsikter och enheter. Formatet kan skrivas antingen som en json-struktur eller i markdown-format som är att föredra då det är kortare och enklare (Bocklisch, Faulkner, Pawlowski, & Nichol, 2017). Exempel på hur det kan se ut är som följande[3]:

```
## intent:greet
- hey
- hello there
- hi
- hello there
## intent:goodbye
- cu
- good by
- cee you later
## intent:mood_unhappy
- my day was horrible
- I am sad
- I don’t feel very well
- I am disappointed
```

Dessa sparas i filer som i sin tur dataprogrammet går igenom och tränar på. Om en användare sedan testar att skriva in en text, exempelvis ”I am unhappy” så skulle datat för hur den klassificerar denna input kunna se ut på följande sätt (skrivet i json struktur)[4]:

[3] Citerat från Pandey (2019)

[4] Citerat från Pandey (2019) med några modifieringar i form av att delar av kod tagits bort. De exempel på avsikter (*eng. intents*) som inte skrivits tidigare kan hittas i artikeln som detta citeras från.

```
{
 “intent”: {
  “name”: “mood_unhappy”,
  “confidence”: 0.7358779973655818
 },
 “entities”: [],
 “intent_ranking”: [
  {
   “name”: “mood_unhappy”,
   “confidence”: 0.7358779973655818
  },
  {
   “name”: “goodbye”,
   “confidence”: 0.03866425150932923
  },
  {
   “name”: “greet”,
   “confidence”: 0.03327667891984685
  }
 ],
 “text”: “I am unhappy”
}
```


Utifrån detta går det att se att programmet har klassificerat och tolkat input korrekt. För att få ytterligare information kring denna kod och hur det fungerar, se Pandey (2019) samt Bocklisch et al. (2017) eller hemsidan för Rasa[5].

2.3 Chatbotar och dess funktionalitet idag

Bayan Abu Shawar och Eric Atwell (Shawar & Atwell, 2007) beskriver chatbotar som konversationsagenter som via naturligt språk interagerar med användare genom turtagning. Vanliga sätt att kunna interagera och konversera med en chatbot antingen via text, interaktiva kort och via tal. Målet är att interaktionen ska ske lika lätt och smidigt som om det vore med en människa, eller åtminstone en intelligent robot. Chatbotar går att implementera på många

[5] https://rasa.com/

ställen, men där de är mest lämpade är vid enkla repetitiva uppgifter som går att automatisera (Iqbal, Berry, Spacil, Qian, & Standefer, 2019). Med repetitiva menas alltså återkommande uppgifter, eller exempelvis frågor, där en chatbot snabbt skulle kunna hantera uppgiften eller svara på dessa typer av frågor. Detta underlättar och minskar eventuell arbetsbelastning på människor och överlåter dessa simplare uppgifter och frågor till en intelligent robot som alltid finns tillgänglig. Vanliga typer av uppgifter är objektiva och innefattar bland annat att leta efter information eller fylla andra administrativa behov (Zamora, 2017). Iqbal et al. (2019) nämner att en chatbots interaktion kan vara enbart fråga/svar men även en sofistikerad konversation där chatboten ger användaren tillgång till olika tjänster.

Dock som Grudin och Jacques (2019) nämner i sin artikel så är det möjligt att dela in chatbotar i tre olika typer; virtuella följeslagare, intelligenta assistenter och uppgiftsfokuserade chatbotar. Denna indelning grundar sig i hur djupt och brett fokus chatboten har. Där brett i princip handlar om hur många olika områden chatboten kan hålla en konversation i, och djupt handlar om hur djupt konversationen kan gå inom det specifika området eller områdena. Virtuell följeslagare har både ett brett och djupt fokus och kan därför hålla en lång konversation igång om vad som helst. ELIZA är, och ALICE antas också vara, en chatbot som är ett exempel på en virtuell följeslagare. Drömmen med Artificiell Intelligens (AI) var när den grundades att skapa så kallad generell intelligens,

> ”…en intelligens för att styra dem alla.” (Grudin & Jacques, 2019)

Virtuella följeslagare tar stegen mot denna dröm. De nästkommande två typerna av chatbotar är något som det legat allt mer fokus på de senaste åren där båda typerna inte har ett lika djupt fokus utan försöker att hålla konversationerna korta. Skillnaden är att intelligenta assistenter, likt virtuella följeslagare, har ett brett fokus och fungerar därmed i öppna domäner. Exempel på denna typ av chatbot är bland annat Siri (Grudin & Jacques, 2019).

Uppgiftsfokuserade chatbotar har istället, som även kan antas på namnet, ett smalt fokus och fungerar enbart i stängda domäner där konversationen enbart handlar om det område som chatboten är specialiserad på. Under denna kategori hamnar exempelvis kundtjänstchatbotar. De senaste par åren har flera hundratals chatbotar, om inte fler, byggts och översvämmat nätet. Uppgiftsfokuserade chatbotar blev under 2016 en mittpunkt av maskinintelligens, men att skapa lyckade sådana visade sig vara svårare än förväntat. Trots att uppgiften kan anses vara enkel så finns det vissa begränsningar i AI systemen för att hantera dessa om de inte kan hantera 100%

av händelser som det står inför. Står den framför en situation den inte är programmerad att hantera blir den snabbt mindre effektiv än en människa. Exempelvis skulle detta kunna vara om chatboten får en fråga om ett relaterat område som den inte är programmerad att kunna svara på men som en mänsklig expert hade kunnat svara på. Detta kallas för en *uncanny cliff* och handlar alltså om att det finns en oväntad skarp gräns mellan vad som är känt och inte. Ytterligare svårigheter som finns med dessa AI-system är att trots att det finns en sådan stor kunskap kring naturligt språk så är den fortfarande inte tillräcklig för att de ska kunna konversera som människor. Den mänskliga konversationen är inte linjär och är i många fall beroende av den fysiska kontexten i form av gester, miner, toner och så vidare (Grudin & Jacques, 2019). Därmed, trots att det finns väldigt mycket kunskap kring chatbot- och AI-tekniken och just utvecklingen av chatbotar så finns det en del begränsningar inom detta forskningsområde.

Io och Lee (2018) genomförde en bibliometrisk analys över hur forskningsområdet kring chatbotar och konversationsagenter har sett ut. Då AI är ett område som växt för varje år som går, samt att folk investerar allt mer i det så anser de att det är väldigt troligt att chatbotar kommer att växa och utvecklas ännu mer i framtiden. De upplever alltså att det finns stora möjligheter med chatbotar och att det är en teknik som kommer att utvecklas mer och mer.

Ett av resultaten de fick utifrån sin analys var bland annat att forskningen kring chatbotar minskade efter 2010. Detta diskuterade de att det skulle kunna bero på antingen den finansiella krisen eller att teknologin inte riktigt fanns där för att kunna vidareutveckla chatbotar och konversationsagenter (CA). Dock efter 2015 såg de istället en ökning, vilket skulle kunna ha en koppling till utvecklingen av AI och relaterad teknologi och därmed alltså att vidareutveckling var möjlig igen. Deras undersökning visade dock även att mycket av den existerande forskningen som är publicerad efter 2015 kring chatbotar och CA inte har applicerat den nya teknologin kring *Deep Learning* (DP) (Io & Lee, 2018). DP har sedan 2011 bidragit till att området AI har gjort stora framsteg där DP-applikationer gör det möjligt att få tillgång till stora mängder av data. Denna teknik använder sig av flerskiktade nätverk med artificiella neuroner som via algoritmer lär sig hur de ska svara för att förbättra klassificeringen (Dunn, 2019). Som tidigare nämnts används dock vanligtvis NLP, tillsammans med maskininlärning, idag vilket är ett område som fortfarande utvecklas. Dock i och med att DP tagits fram bör detta område utforskas mer anser Io och Lee (2018). På grund av denna utveckling av både NLP och DP finns det stor potential i att utveckla och öka forskningen kring chatbotar och CA.

De upptäckte även i sin analys att det finns en brist på akademisk forskning speciellt inom verksamheter, och speciellt kring just industriell användning av chatbotar. De anser bland annat att forskning bör utgå mer ifrån hur verksamheter kan gynnas från användande och implementering av chatbotar. Forskningen bör även fokusera på vad för roll chatbotar kommer ha, hur användningen av chatbotar kan påverka människans attityder samt andra relaterade problem (Io & Lee, 2018).

Undersökningen som Io och Lee (Io & Lee, 2018) genomfört har visat på att det finns en stor bredd i forskningen på chatbotar, alltså att de undersökts inom flera olika typer av områden. Detta kan anses vara både positivt och negativt. Positivt då det är många områden som påbörjat att utforskas och är därmed starten på mer kunskap, men negativt bland annat för att forskningen har en bristande styrka i sig. Det krävs mer forskning för att eventuellt styrka fynden som gjorts. Dessutom kan det vara en bidragande faktor till ett annat fynd de gjorde, vilket var att de inte kunde hitta nyckelord som *mobile*, *Deep Learning* eller *Business Intelligence* kopplade till detta forskningsområde. Vilket de anser implicerar att det finns en stor forskningsmöjlighet att utforska dessa ytterligare i koppling till chatbotar och CA.

Inte nog med att det upplevs finnas vissa begränsningar i forskningsområdet kring chatbotar, så uppmärksammar Jennifer Zamora (2017) även bristande kunskap i att definiera dess syfte och det värde som de kan erbjuda. För att kunna bedöma om en chatbot är av hög kvalitet behöver den möta de förväntningar användarna har på den. Möter inte chatboten användarnas förväntningar finns det en stor risk att de inte använder den. En annan orsak till att en chatbot inte används är ifall uppgiften upplevs som mer komplex, jobbig eller på något annat sätt sämre när chatboten används än att utföra uppgiften utan den.

Förväntningar är svåra att möta då dessa kan variera väldigt mycket från person till person då alla individer har olika erfarenhet och standard på andra tjänster. Utifrån den undersökning som Zamora (2017) genomförde visade det sig att en vanlig sökmotor möjligtvis hade varit mer effektiv och informativ än en chatbot vid informationssökning och mer komplexa uppgifter. Anledningen till detta kan vara att det redan finns en stor vana och kunskap just inom sökmotorer. Dock i och med chatbotars utvecklingsmöjligheter och växande popularitet bör detta vända.

2.4 Chatbotens uppbyggnad och design

En chatbots livscykel, se bild 1, består av flera olika moment där majoriteten av momenten görs om flertalet gånger. Första steget i denna process är att planera och designa chatboten, det är även detta steg som fokuset ligger på i det här projektet. För att skapa en framgångsrik chatbot krävs det att det finns en bra förståelse för målen, processerna och användarbehoven för den typ av chatbot som ska skapas.

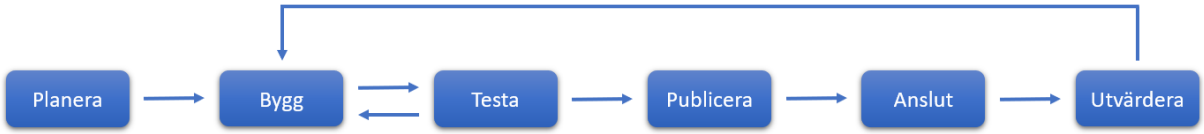


Bild 1. Illustrering av en chatbots livscykel.

Den högsta prioriteten vid skapande av en chatbot bör ligga på att skapa en bra användarupplevelse. För att vara säker på att uppnå detta kan det vara bra att fundera kring dessa frågor som är citerade från Microsofts designriktlinjer (Velloso, Iqbal, & Standefer, 2017b) (egen översättning):

- Löser chatboten enkelt användarens problem med minsta möjliga antal steg?
- Löser chatboten användarens problem bättre/enklare/snabbare än något av de alternativa upplevelserna?
- Körs chatboten på enheterna och plattformarna som användaren bryr sig om?
- Är chatboten upptäckbar? Är det självklart för användarna vad de ska göra när de använder den?

Mer detaljer kring hur en design av en chatbot ska göras samt vad det finns för riktlinjer kommer att redogöras nedan, under rubriken Designriktlinjer.

Nästa steg i processen är att bygga samt implementera chatboten och den design som tagits fram. För att bygga chatbotar går det bland annat att bygga de från grunden själv, eller så går det att använda sig av tjänster som innehåller olika typer av ramverk. Bland annat finns det Microsoft Bot Framework (Azure), IBM Watson Chatbots med flera (Biswas, 2018).

En chatbot kan beskrivas som en modern webapplikation som använder sig av olika API (*Application Programming Interface*) för att skicka och ta emot meddelanden (Iqbal et al., 2019). API är ett mjukvaru-till-mjukvaru-gränssnitt som hjälper utvecklare som bygger

applikationer att exponera olika typer av tillgångar och data. Ett API definierar ett så kallat kontrakt som beskriver hur applikationer kan interagera med varandra via ett nätverk utan någon mellanliggande användarinteraktion (De, 2017).

Chatbotar är uppbyggda på olika sätt och innehåller olika typer av programvaror och kod beroende på vad chatboten ska prestera. De skulle kunna läsa och skriva i filer, använda databaser och API's med mera, precis som annan typ av mjukvara kan göra. Vad som skiljer chatbotar åt från vad andra typer av mjukvaror kan göra är just naturligt språk-interaktionerna (Iqbal et al., 2019).

Det är alltså under detta steg, bygg- och implementationssteget, som detta implementeras utifrån vad som behövs och vad chatboten ska användas till. Därefter behöver chatboten testas innan den ska publiceras. Genom att testa chatboten kan buggar och eventuella felaktiga beteenden hittas och därmed göra det möjligt att ändra på detta. Därav sker en iteration där chatboten byggs, för att sedan testas och eventuellt byggas på igen som i sin tur leder till att den behöver testas igen. Detta genomförs fram tills att ett önskvärt resultat har uppnåtts, vilket leder till nästkommande steg som är att publicera chatboten. Detta gör chatboten tillgänglig för användning och är ett första steg för att chatboten ska komma till liv (Iqbal et al., 2019).

Det näst sista steget handlar om att ansluta chatboten till en eller flera kommunikationskanaler, som exempelvis Facebook, Skype, Cortana eller liknande. Dock är detta steg inte av relevans för just detta projekt och uppdrag.

Slutligen ska chatboten utvärderas, detta kan göras genom att samla in data på olika sätt för att ta reda på hur mycket den används, om den ger de svar som användarna förväntar sig och så vidare. Utifrån denna utvärdering kan det vara så att chatboten behöver byggas om eller ändras på. Detta leder till att processen går tillbaka till steg två igen och därefter fortsätter att gå steg för steg tills att processen är på sista steget igen. Denna iteration bör ske kontinuerligt för att hålla chatboten uppdaterad.

2.4.1 Designriktlinjer

Microsoft har tagit fram tio riktlinjer som riktar sig till utvecklare av chatbotar (Microsoft, 2018). Dessa riktlinjer fokuserar, enligt dem, på att hjälpa utvecklaren att designa en chatbot som ger användaren tillit till företaget och tjänsten som den levererar. De menar även på att dessa riktlinjer är mest relevant för de chatbotar som påverkar människor på ett konsekventiellt sätt. Med detta menas alltså att chatboten har en signifikant påverkan på användarens dagliga

liv. Utifrån detta projekt är det, som tidigare nämnts, tänkt att chatboten ska underlätta supportteknikernas arbete och därmed bör den ha en signifikant påverkan på dem. Skulle chatboten inte kunna hjälpa användaren skulle detta alltså bidra till en meningsfull påverkan på individen.

> 1. Articulate the purpose of your bot and take special care if your bot will support consequential use cases.

Detta är den första riktlinjen som tas upp, och handlar om att vara tydlig med vad syftet med chatboten är och hur den hanterar konsekventiella användningsfall. För att uppnå denna riktlinje har de beskrivit tre punkter att tänka på. Första punkten handlar om att vara noggrann med att innan designarbetet påbörjas måste man specificera hur den chatbot som ska utvecklas ska gagna användaren (Microsoft, 2018). Hur kommer chatboten påverka användarens liv? Chatbotar måste vara effektiva och spara tid för att människor ska vilja använda dem, de ska enbart hantera simpla uppgifter. Dessa simpla uppgifter ska vara relaterade till dagliga rutiner, dock kan de fortfarande resultera i både låga och höga konsekvenser vid misslyckande. Ska känsliga uppgifter hanteras bör detta ses över ordentligt så att chatboten kan hantera detta och får tillräckligt med tillit från användaren innan utvecklingen av den påbörjas (Zamora, 2017). Detta är essentiellt för en bra användarupplevelse.

Därefter måste en bedömning göras för ifall det avsedda syftet med chatboten kan utföras av den på ett ansvarsfullt sätt, vilket är punk nummer två. Finns all relevant expertis inom domänen som chatboten ska operera och kommer den kunna svara så pass korrekt att det inte leder till några större konsekvenser? Det här leder till den sista punkten som handlar om att göra det möjligt att utvärdera användarupplevelsen av chatboten. Detta för att inte bara få reda på ifall chatboten gör det den ska utan även få reda på hur användaren känner vid användningen av den (Microsoft, 2018).

> 2. Be transparent about the fact that you use bots as part of your product or service.

Den andra riktlinjen handlar mer om användarens tillit till företaget som använder sig av chatbot-teknologi samt även tillit till själva chatboten. För att uppnå detta krävs det att det finns en tydlighet mot användaren kring dels att de vet att de interagerar med en chatbot och inte en annan person samt dels med vad chatboten kan och inte kan göra (Amershi et al., 2019; Microsoft, 2018). Anledningen till att det bör vara tydligt att det är en chatbot de skriver med kan bero på att det inte ska uppstå en frustration eller förvirring i att chatboten inte helt kanske

agerar som en människa skulle gjort. Dock menar Zamora (2017) på att det finns en viss fördel med att chatboten har människoliknande egenskaper då det finns en viss misstro till datorbaserade system. Så utifrån detta krävs det nog att en avvägning görs på hur viktigt det är i detta sammanhang och för denna tjänst att det finns människoliknande egenskaper. Vad gäller tydlighet kring vad chatboten kan och inte kan göra handlar om att det bland annat ska vara tydligt för användaren vad chatboten är till för. Dessutom ska det även vara tydligt vad för eventuella fel den kan göra och vad detta kan leda till för konsekvenser (Amershi et al., 2019; Microsoft, 2018). Det bör även finnas möjlighet för användaren att hitta mer information om chatboten (Microsoft, 2018). Är det otydligt för användaren vad chatboten kan och inte kan göra så kan detta begränsa användarna i att få den att fungera som den är tänkt. De kan bland annat anta att den funktionalitet som finns är väldigt begränsande eller att de bland annat blir överväldigade av dess okända potential (Luger & Sellen, 2016).

Första intrycket som användaren får kan vara avgörande för denna riktlinje. Detta eftersom det är på första sidan, eller första meddelandet, som bidrar till att användarna kan få en direkt förståelse för hur chatboten ska användas, vart de kan få hjälp och så vidare. Därför är det viktigt att i designprocessen ta till vara på hur användarens första intryck blir (Velloso, Iqbal, & Standefer, 2017a).

> 3. Ensure a seamless hand-off to a human where the human-bot exchange leads to interactions that exceed the bot's competence.

Denna riktlinje handlar om att det bör finnas ett simpelt sätt som ärendet eller konversationen kan överlämnas på till en mänsklig moderator ifall att chatboten inte kan hantera det. Detta bör göras så fort användaren ber om det, eller om användaren är missnöjd vilket bör indikeras av användaren eller upptäckas av chatboten. Görs inte detta finns risk för att en misstro till teknologin skapas (Microsoft, 2018).

> 4. Design your bot so that it respects relevant cultural norms and guards against misuse.

Som nämnts tidigare är det viktigt att chatboten beter sig på ett sätt som användaren förväntar sig att den ska göra. Dessa förväntningar kan dock skilja sig åt beroende på sociala och kulturella kontexter (Amershi et al., 2019). Fjärde riktlinjen går in på hur chatboten bör designas för att interaktionen ska ske på ett respektfullt sätt gällande kulturella normer, där det inte ska ske några kränkningar eller aggressiva beteenden. Detta görs genom att bland annat begränsa chatboten så gott det går utifrån det den syftar till att göra. Dessutom kan exempelvis

maskininlärningstekniker eller mekanismer för nyckelordsfiltrering användas för att chatboten ska upptäcka opassande beteende och kunna svara därefter (Microsoft, 2018).

5. Ensure your bot is reliable.

Nästa riktlinje fokuserar på chatbotens trovärdighet från olika perspektiv. Bland annat är det viktigt att tänka på att kontinuerligt kontrollera att chatboten svarar som den är tänkt att svara. Trovärdighetssignaler bör då utvecklas hos chatboten för att den ska ha möjlighet att fastställa när den gjort fel och meddela detta till användaren och/eller skicka vidare det till en människa som kan ta över ärendet. Det är därför även bra, som tidigare nämnts, att vara säker på att kunskapen som erhålls inom domänen chatboten ska användas i grundar sig i expertkunskaper. Även i detta fall är det viktigt att det finns en transparens i chatbotens trovärdighet, alltså att det finns tydlig information kring chatbotens presentation exempelvis i form av sammanfattningar av generell statistisk prestation eller liknande (Microsoft, 2018).

För att ge en trygghet till användaren och en säkerhet som utvecklare bör det vara möjligt för användaren att lämna feedback. Denna feedback bör chatboten aktivt fråga efter och även vara tydlig med om den kommer ge användaren något svar på det eller inte (Amershi et al., 2019; Microsoft, 2018). En annan faktor som skulle kunna öka tryggheten är att skapa en spårbarhet, vilket alltså gör det möjligt att gå tillbaka och se eventuellt vart det blev fel. På grund av att det är ett AI-system finns det alltid en risk att det kommer förse med felaktiga svar då dessa system är probabilistiska (Microsoft, 2018).

6. Ensure your bot treats people fairly.

Den sjätte riktlinjen handlar om att se till att chatboten behandlar användarna rättvist (Amershi et al., 2019; Microsoft, 2018). Detta kan göras genom att kontinuerligt se till att den data som används för att träna chatboten har en lämplig representativitet och kvalitet. Dessutom kan det vara bra att utvecklingsteamet för chatboten har en mångfald då detta kan leda till att chatboten i sin tur agerar mer rättvist (Microsoft, 2018).

7. Ensure your bot respects user privacy.

Denna riktlinje har ett fokus på användarinformation och hur denna ska hanteras. När människor interagerar med exempelvis en chatbot kan det i vissa fall hända att de delar med sig av mer information än de skulle gjort om det var en människa. Vilket blir problematiskt när en chatbot troligtvis minns mer än vad en människa gör då de kan minnas allt. Därför är det oerhört viktigt

att respektera användarens privata information. För att göra detta är det viktigt att vara tydlig med användaren om vad för uppgifter som sparas, vad det ska användas till och vad för sekretesspolicy som finns. Information kring detta bör även vara tydligt för användaren var den finns någonstans. Den informationen som i sin tur samlas in ska inte heller vara mer än det nödvändigaste och informationen bör över tid även rensas för att minska säkerhetsrisker för användarna. I vissa fall kan det även vara nödvändigt att göra det möjligt för användaren att dels se vad för information som sparats samt dels radera viss information. Utifrån detta kan det därför också vara bra att få både en juridisk- och sekretessbedömning av den sekretesspraxis som finns för den chatbot som ska utvecklas (Microsoft, 2018).

8. Ensure your bot handles data securely.

Den åttonde riktlinjen har en stark koppling till föregående riktlinje och handlar om att och hur data hanteras säkert. Det är otroligt viktigt att det finns säkra mjukvaruunderlag, ordentlig autentisering, inputvalidering och så vidare för att inkräktare inte ska komma åt känsliga data. En annan säkerhetsåtgärd som kan vara bra att implementera är att chatboten ska kunna upptäcka onormalt beteende och förhindra försök till manipulering genom att de exempelvis försöker trigga delar av chatbotens programmeringskod. Detta kan bli svårt, men väsentligt, för chatboten att kunna skilja på ifall den data som kommer in enbart är ovanlig eller om den på något vis försöker komma åt känslig information. Om det finns ett lämpligt säkerhetsteam inom organisationen som ska utveckla en chatbot kan det vara lämpligt att använda detta team för att göra en säkerhetsgranskning på chatboten (Microsoft, 2018).

9. Ensure your chatbot is accessible.

Näst sista riktlinjen inkluderar och värnar om att alla användare, även de med olika typer av funktionsnedsättningar, ska kunna använda chatboten. För att kunna uppnå detta krävs det av chatboten att den har viss funktionalitet för att kunna hantera användare som exempelvis enbart använder tangentbordet för att navigera sig, eller kräver att det finns färgkontraster eller skärmläsare och så vidare. Därav kan det vara viktigt att tänka på detta vid designarbetet av chatboten samt att även testa med personer som har olika typer av nedsättningar för att vara säker att gränssnittet passar alla typer av användare (Microsoft, 2018).

10. Accept responsibility.

På grund av begränsningarna som finns idag vad gäller chatbotars möjligheter att fullt ut agera autonomt så är den sista riktlinjen skapad. De som utvecklar chatboten är ansvarig för vad den gör och hur den påverkar folk. Skulle det dock vara en tredje part som skulle utveckla denna chatbot måste en överenskommelse göras över vem som är ansvarig. På grund av detta är det viktigt att se över så att den chatbot som organisationen utvecklat följer dessa riktlinjer (Microsoft, 2018).

Zamora (2017) diskuterar även kring ytterligare faktorer som påverkar användarupplevelsen, bland annat språket som används. Det är fördelaktigt att hålla sig till ett språk då det skapar stor problematik om språk ska blandas. Denna problematik har att göra med begränsningar i språkbearbetningstekniken som finns idag. Det är även viktigt att fundera kring vilket språk som ska användas för att användaren både ska vara bekväm och kunna uttrycka sig på ett korrekt sätt.

En annan faktor som diskuterades var inputmodaliteter och vilken typ som bör användas i vilka situationer. För mer komplexa uppgifter är det bättre att använda sig av text då det i sådana fall blir lättare att formulera sig. Tal kan vara fördelaktigt om användaren utför flera uppgifter samtidigt och därmed har händer och/eller ögon upptagna på annat. För att förbättra användarupplevelsen kan därmed dessa två komplettera varandra genom att erbjuda flera inputkanaler. Dock är det helt kontextberoende för vad som passar i vilket situation, därför är det viktigt att ha detta i åtanke vid design av inputmöjligheter (Zamora, 2017).

3. Metod

För detta projekt gjordes inledande datainsamling i forma av kontextuella intervjuer som sedan användes för att skapa ett affinitetsdiagram. Denna initiala undersökning beskrivs i detalj i arbetet skrivet av Linn Olssons (2019). Inledningsvis ges en kort beskrivning av hur affinitetsdiagrammet skapades. Därefter beskrivs det hur kravformuleringen genomfördes och vad för metoder som användes för att genomföra detta. Till sist presenteras den designmetod och det implementeringsverktyg som användes.

3.1 Kontextuella intervjuer och affinitetsdiagram

För att samla in data genomfördes kontextuella intervjuer för att få en uppfattning om hur supportteknikerna arbetar idag. De kontextuella intervjuerna utfördes i enlighet med kontextuell design där forskaren tar en novisroll under ledning av experten, i detta fall en supporttekniker (Holtzblatt & Beyer, 2017). Hur dessa kontextuella intervjuer genomfördes kan läsas om i Linn Olssons uppsats (2019). Dessa intervjuer var i sin tur grunden till det affinitetsdiagram som skapades.

Affinitetsdiagram används för att organisera ostrukturerad data och avslöja vanliga problem och teman för samtliga användare (Holtzblatt & Beyer, 2017). Det är ett väldigt simpelt och vanligt sätt att analysera kvalitativa data på och är ett lämpligt verktyg att använda vid designprojekt (Arvola, 2019; Holtzblatt & Beyer, 2017).

För att skapa ett affinitetsdiagram skrivs inledningsvis data från intervjuerna ner på post-it-lappar. Sedan randomiseras dessa lappar för att placeras ut och grupperas där dessa grupper ska vara ganska små och enbart beskriva ett problem eller en synpunkt. Genom att hålla grupperna små är det större chans att finna fler problem och insikter. För att genomföra detta börjar lapparna att sättas upp så att designteamet kan jämföra de lappar som sitter på tavlan med de lappar som de håller. Verkar lapparna höra ihop på ett eller annat sätt så sätts de ihop, och tillslut skapar en grupp. Dessa grupper blir var och en tilldelade en blå post-it med en unik kortfattad beskrivning av gruppen. Denna beskrivning ska representera de underliggande lapparna så pass bra att man inte behöver läsa något annat än den blåa lappen (Holtzblatt & Beyer, 2017).

Därefter organiseras dessa in i grupper på liknande sätt men ges istället rosa post-it med en beskrivning över de blå lapparna som grupperats. Till sist grupperas de rosa lapparna in och ges gröna post-it med en beskrivning över de rosa lapparna som grupperats. Detta skapar alltså en

hierarkiskt uppdelning och organisering som gör det lättare att se mönster i de problem och insikter som uppfattats finnas (Holtzblatt & Beyer, 2017). Linn Olsson beskriver denna metod samt hela processen i mer detalj i sitt arbete (Olsson, 2019).

3.2 Kravformulering

För att undersöka var och hur en chatbot kan implementeras i Siemens system för supporttekniker krävdes en undersökning i vad för krav ett sådant system ska ha. Detta genomfördes med hjälp av en kravformulering. I just detta projekt utgick denna metod utifrån den beskrivning Mattias Arvola ger i sitt utkast för sin nya upplaga av Interaktionsdesign och UX (Arvola, 2019). Detta då det är en väletablerad metod för framtagning av designkrav. Bland annat har även Goodwin (2009) givit en beskrivning på denna metod som Arvola (2019) till viss del grundar sin beskrivning på. Dock valdes beskrivningen Arvola ger då den var på svenska samt att Arvola fanns tillgänglig för att tydliggöra eventuella oklarheter och som stöd vid genomförande av kravformuleringen.

Vid kravframtagning är det viktigt att skilja på behov och lösningar, där behov är krav medan lösningar, som säger sig självt, är en möjlig lösning på ett krav. För att vara säker på detta kan frågan ”varför?” ställas, exempelvis varför ska slutresultatet vara webbaserat? Finns det ett svar på detta så är det troligt att detta svar är behovet, eller kravet. Att slutresultatet exempelvis ska vara webbaserat skulle kunna vara en potentiell lösning på ett krav. Det överliggande svaret skulle dock kunna vara att systemet som redan finns kanske är en skrivbordapplikation och på så vis skapar en begränsning i att lösningen måste vara webbaserad. Denna begränsning kan på så vis betraktas som ett krav. Av denna anledning är det som sagt viktigt att separera behov från lösningar (Arvola, 2019).

På grund av detta är det viktigt att kraven inte är för konkreta då det troligtvis skulle luta mer åt att det skulle vara lösningar istället för krav. Dock är det även viktigt att kraven inte är för abstrakta heller. Detta eftersom krav måste vara valideringsbara, alltså att det ska vara möjliga att påstå ifall de är uppfyllda eller inte. Kraven måste även vara entydiga, vilket går ihop med att de ska vara valideringsbara. Är kraven inte entydiga kommer de inte vara valideringsbara (Arvola, 2019). Ett bra krav ska alltså vara:

- *Abstrakt* till den grad att de inte specificerar en viss implementerad lösning.
- *Entydig*, så att det tydligt går att förstå hur kravet uppnås och får därmed inte vara för abstrakt.

- *Valideringsbar*, vilket går ihop med entydigheten då kravet måste kunna uppfyllas.
- *Spårningsbar* till sin källa eftersom kravet kan komma från många olika håll, exempelvis från kunders måluppsättningar, konkurrentanalyser eller användarstudier och personas.

Kraven för detta projekt ska grunda sig i bland annat intervjuerna med supporttekniker på Siemens, men även till stor del av bakgrundslitteraturen. Detta då litteraturen kring chatbotar är viktig för att ställa krav kring hur en chatbot designas på bästa sätt.

Första steget var att skapa en persona för att få fram en tydlig bild av vilka användarna för den produkt som ska skapas är. Personas är även viktig för att ta fram användaranpassade krav. Nästa steg var att skapa kontextscenarion, detta används för att skapa situationer av användningsområden och för att via dessa scenarion kunna extrahera krav och även för att kunna bekräfta att idéerna som finns kring produkten är vettiga. Dessa krav kommer i sin tur att kategoriseras som antingen datakrav, funktionskrav, kvalitetskrav eller begränsningar. Därefter ska krav utifrån andra källor extraheras. När kraven framställts är det sedan dags för att filtrera och prioritera kraven (Goodwin, 2009).

Det är viktigt att förstå att det inte krävs att alla krav är satta eller att förståelsen för problemet är fullständig innan utvecklingen av en lösning påbörjas. Som Arvola (2019) skriver finns det ett inneboende beroende mellan problemformulering och lösningsförslag. Ju mer förståelsen växer fram för vad lösningen ska vara desto bättre blir även bilden av vad problemet är. Det är bättre att sätta upp de viktigaste kraven och påbörja design och implementeringsfasen än att skapa alla krav från början.

3.2.1 Persona

En persona används för att öka sin designempati och hjälper designteamet att få en gemensam bild över vem de designar för, samt att det går att diskutera kring just den specifika karaktären och inte om användarna generellt. De är alltså inget porträtt av en enskild individ utan är ett sätt att gestalta en sammanställning av de tänkta användarna. För att denna metod ska fungera och vara det kraftfulla kommunikationsverktyg som det är, krävs det att personan blir levande. Att en persona blir levande innebär alltså att karaktären inte bara får namn och arbetsbeskrivning utan även en personlighet och en inblick i dennes livssituation. Det är vanligt att det i ett designprojekt skapas flera personas, dock är det enbart en persona som är den primära, alltså

den karaktär som är utgångspunkten. Därefter kan de sekundära personornas mål uppfyllas, så länge de inte motsätter målen den primära personan har (Arvola, 2014).

Det tillvägagångssätt som användes för att skapa personan grundar sig till stor del på Goodwin (2009), men där beskrivningen av Arvola (2014) följs. De första två stegen handlar om att att dela upp de intervjuade i olika roller och om att bekanta sig med den data som samlats in. De nästkommande två steg innefattar att fördjupa sig på individnivå genom att skapa beteendevariabler. Dessa variabler kan skapas utifrån de kategorier som tagits fram i affinitetsdiagrammet, där en kategori kan översättas till en eller flera variabler. Dessa variabler ska även kunna uttryckas i intervaller, exempelvis intervallet från låg till hög. Ett exempel som Arvola (2014) tar upp är att det på ena sidan skulle stå ”söker hjälp från andra” och ”försöker själv” på andra sidan. Därefter ska de individer som intervjuats placeras ut på dessa variabelintervaller. De individer som i sin tur förekommer tillsammans på flera variabler skapar underlaget för en persona. Därefter är nästa steg att ta fram tre till fyra mål för varje persona som handlar om vad de deltagarna som skapade denna persona är ute efter. Det är viktigt att målen också har relevans för problemområdet för just det specifika designarbetet. Skapas det fler än en persona är det viktigt att de görs distinkta samt att de ska prioriteras där en ska vara primär och resten sekundära. Till sist ska personan eller personorna göras mer levande, med hjälp av att ge personan ett ansikte i form av en bild eller illustration, samt att ge den en berättelse och övrig kommunikation runt personan.

Avsteg från denna metod gjordes genom att ingen fördjupning på individnivå genomfördes. Detta eftersom det vid datainsamlingen och skapande av affinitetsdiagrammet inte gjordes någon skillnad på individerna. Individerna ansågs vara tillräckligt lika för att kunna skapa en enda persona.

3.2.2 Kontextscenario

Kontextscenarion är en metod som kan användas, och har använts i detta projekt, för att skapa krav. Med hjälp av denna metod kan produktens ideala framtida beteende och hur den ska möta personans behov beskrivas(Arvola, 2019; Goodwin, 2009). Under denna process är det viktigt att ta fram barnet inom sig som ser möjligheter i allt för att få fram många idéer och tankar kring berättandet som ska infinnas i kontextscenariot (Goodwin, 2009). Det första steget är att identifiera vilka kontextscenarion som ska vara med, vilket gjordes genom en brainstorming-session. Brainstorming är en metod som kan användas för att få fram så många idéer på så kort tid som möjligt. Här ska man tänka vitt och brett, där inga idéer är dumma idéer. För detta

projekt avsattes 10 minuter för att ta fram så många idéer som möjligt. Viktiga punkter att tänka på vid skapande av kontextscenarion är bland annat att scenariot ska vara skrivet från personans synvinkel samt att scenariot, eller berättelsen, har både en start och ett slut (Arvola, 2019; Goodwin, 2009). Att scenariot är från personans synvinkel innebär alltså att enbart det som är synligt för användaren ska beskrivas, alltså inte bakomliggande funktionalitet inom system. Detta eftersom denna metod fokuserar på design av användargränssnitt. Det är även bra att ta med eventuella känslor och motivationer i scenariot då detta också är en del av personans synvinkel (Goodwin, 2009).

Det är som sagt även viktigt att få med hela scenariot, från start till slut. Scenariot kan vara en uppgift som består av en eller flera funktioner som tar några få minuter eller flera timmar och som kan innehålla ett eller flera avbrott på grund av olika orsaker. Detta är viktigt att få med för att hela händelseförloppet ska vara med. Ett kontextscenario består ofta av en utlösande händelse, beskrivning av informationsutbyte mellan persona och system, beslut som personan tar och handlingar som den utför och slutligen resultatet av detta som personan ser (Goodwin, 2009).

Varje kontextscenarios som väljs ut bör innehålla åtminstone ett exempel på en grundläggande aktivitet som de personas som skapats troligtvis ska utföra. Är aktiviteterna kopplade till varandra där ena aktiviteten sker i följd av den andra bör de vara i ett och samma scenario. Exempelvis om personan har redaktör som yrkesroll skulle ett kontextscenario kunna vara Publicera annonser (Arvola, 2019).

Ett kontextscenario skrivs därefter ut i löpande text som en berättelse. När detta ska göras är det bra att tänka på och besvara frågorna vem gör vad, när, var, hur och varför (Arvola, 2019). För detta projekt skapades enbart ett kontextscenario då flera ansågs vara överflödiga då produktens uppgift var väldigt begränsad.

3.2.3 Kategorisering av krav

När en uppsättning av kontextscenarion har skapats är det dags att extrahera kraven och skriva upp dessa i listor. Det är även viktigt att inte glömma eventuella krav från andra källor exempelvis via konkurrentanalyser eller från intervjuer med beställare. Här är det viktigt att tänka på de fyra punkterna som tidigare nämnts för att skapa bra krav, alltså att de ska vara abstrakta, men entydiga, valideringsbara och spårbara. Spårbarheten ska alltså visa kravets ursprung, exempelvis från vilket scenario och vilken persona det kommer från (Arvola, 2019).

För att se till att alla aspekter har täckts kan det vara bra att kategorisera kraven. Detta kan göras på många olika sätt med många olika typer av metoder och kategoriseringar. Det finns bland annat en metod framtagen av Carlshamre (Carlshamre, 2001) som delar upp kraven som funktionella eller icke-funktionella. En problematik med denna metod är att det inte finns några vedertagna kategoriseringar av just icke-funktionella krav. Den modell som används i detta projekt för skapande av krav är en modell framtagen av Goodwin (2009), som inte har detta problem och som består av fyra kategorier:[6]

- Datakrav – de delar av information som det interaktiva systemet ska innehålla. För att finna dessa krav är ett första steg att leta efter substantiv i den data som samlats in, vilket kan vara allt från intressentintervjuer till personas.
- Funktionskrav – det användaren ska kunna göra med och i det interaktiva systemet. Det är alltså de verb som talar om vad systemet ska göra. Dock innebär enbart de funktionella kraven att systemet fungerar rent tekniskt men inte nödvändigtvis praktiskt.
- Kvalitetskrav – de egenskaper eller upplevelsemål som det interaktiva systemet ska ha eller uppnå. Egenskaperna är kvantifierbara och pragmatiska och kan vara allt från färg och form till ett systems prestanda, systemsäkerhet och responstid. Upplevelsemålen är istället grundade i emotionella kvalitéer och beskriver vad användaren ska uppleva.
- Begränsningar – externa faktorer som påverkar slutresultatet av produkten. Dessa faktorer kan exempelvis vara en viss tidpunkt när produkten måste vara klar, att utformningen av systemet måste följa vissa lagar eller förordningar eller att en viss typ av teknisk plattform måste användas. Här är det även viktigt att skilja på vilka begränsningar som är fasta och vilka som är flexibla för att se vilka som faktiskt är krav och vilka som enbart bör tas hänsyn till.

3.2.4 Prioritera krav

Prioriteringar av de krav som tagits fram kan göras på olika sätt, bland annat tillsammans med eller av intressenter genom att de får jämföra krav parvis och försöka bedöma hur mycket viktigare det ena kravet är än det andra utifrån en relativ skala. Dock tenderar olika intressenter

[6] Dessa kategorier beskrivs både utifrån Arvola (Arvola, 2019) och Goodwin (2009)

att ha olika agendor därför är det bra att reflektera över vems fokus och agenda som är viktigast för just detta projekt. Det skulle även gå att göra med flera parter för att kunna stärka prioriteringen men där eventuellt fokus på enbart en bör finnas för att prioriteringen ska passa projektet på bästa sätt (Arvola, 2019).

Ett annat sätt är att poängsätta de olika kraven med något poäng mellan 1–5. Dessa poängsättningar görs för tre olika faktorer, värdet för kunden, värdet för företaget och enkelheten att implementera. Det sammanlagda värdet av dessa poäng för varje enskilt krav skapar prioriteringen, den med högst poäng prioriteras högst och så vidare (Ratcliffe & McNeill, 2012).

Då fokuset för just detta projekt ligger i supportteknikerna och deras perspektiv på produkten ansågs deras åsikter i prioritering av krav som väldigt viktiga. Prioriteringsmetoden som användes var en modifierad variant av att poängsätta kraven. Eftersom det var supportteknikerna som skulle prioritera är det till störst det värdet för kunden som är av störst relevans, då produkten ska fungera som ett hjälpmedel för dem. Ett formulär skapades med hjälp av Google Forms där kraven de skulle prioritera samlades och där det fanns möjlighet att poängsätta alla dessa krav mellan 1–5. Detta formulär mejlades sedan ut, tillsammans med en beskrivning över hur de skulle gå tillväga, till alla supportteknikerna. Att använda poängsättning istället för att de skulle prioritera var för att underlätta för supportteknikerna att svara på formuläret. Att poängsätta kraven istället för att försöka prioritera dem ansågs ta kortare tid och därmed troligtvis öka chansen för att supportteknikerna skulle svara. Därefter sammanställdes poängsättningen på alla krav och kraven prioriterades utefter detta.

3.3 Skissning av prototyp

De prioriterade kraven samt kontextscenariot var grunden för designprocessen för den prototyp som skulle framställas. För att påbörja designen av prototypen genomfördes gränssnittsskissning (Arvola, 2014). Vid gränssnittskissning skissas mekanismer, strukturer, komponenter och även helheten av den slutgiltiga produkten. Olika typer av tecken används för att förtydliga designbeslut, delar att utforska mer eller värdera de olika alternativen. Plus-minus-listor används för att värdera alternativ, där beskrivs positiva och negativa aspekter av alternativet för att underlätta val av designalternativ. Frågetecken används för att markera ifall någon designskiss bör utredas vidare. Slutligen används utropstecken för att markera designbeslut (Arvola, 2014).

3.4 Prototypningsverktyg

För att skapa en prototyp användes verktyget Adobe XD[7]. Detta verktyg användes då det ansågs vara ett simpelt verktyg att använda för att kunna framhäva de funktioner som chatboten skulle ha i detta projekt. Det finns prototypningsverktyg för att just skapa chatbotar, dock ansågs de som undersöktes inte vara lämpliga för att visualisera just den funktionalitet som denna chatbot skulle ha. Med prototypningsverktyg specialiserade för chatbotar blir man låst till specifika utseenden. Därför ansågs ett prototypningsverktyg som exempelvis Adobe XD vara bättre för just detta projekt.

[7] https://www.adobe.com/se/products/xd.html

4. Resultat

I detta avsnitt beskrivs först och främst de delar av affinitetsdiagrammet som var av relevans för detta projekt. Därefter presenteras personan och det kontextscenario som kunde framställas utifrån denna data. Sedan beskrivs de krav som skapats, hur de poängsattes av supportteknikerna och hur prioriteringen av dessa krav blev. Slutligen demonstreras den slutgiltiga prototypen, där även de designbeslut som tagits blir beskrivna.

4.1 Affinitetsdiagram

Det som kommer presenteras i detta avsnitt är de delar ur affinitetsdiagrammet som ansetts vara relevanta för detta projekt och som har haft en inverkan på resultaten för personan, kontextscenariot och kraven. De gröna rutorna är den högst rankade gruppen, därefter kommer de rosa lapparna och sist de blå.

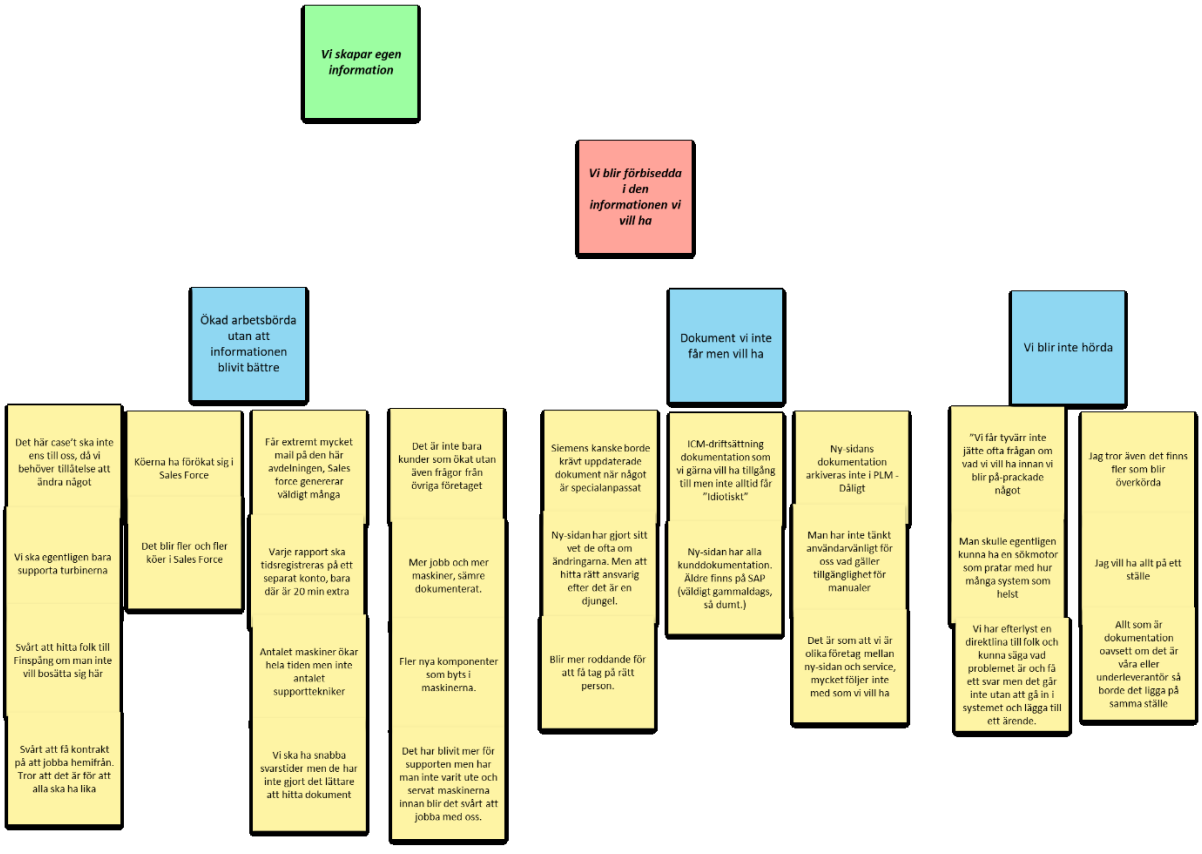


Bild 2. Del av affinitetsdiagrammet som beskriver missnöjet i att supportteknikerna inte får den information de vill ha.

Bilden ovan visar en del av affinitetsdiagrammet som handlar om att de skapar egen information då känner att de inte blir hörda vad gäller vilken information de vill ha. Denna del av affinitetsdiagrammet förmedlar hur arbetsbördan har ökat för supportteknikerna men att

systemen och informationen de ska använda sig av inte har förbättrats. Det finns dokumentation de skulle vilja ha men som de inte får och de får inte de typer av system de skulle vilja ha och använda sig av.

Nästa bild, se bild 3, tar upp att det finns en stor säkerhet kring hantering av dokument där det krävs en viss behörighet och inloggning för att se vissa dokument. Sedan tar denna även upp viss problematik kring att det kan ta tid att lösa kundproblem då de på supporten behöver komma i kontakt med annan personal som har expertkunskap inom det specifika området. Detta medför att det kan bli en del väntande på svar.

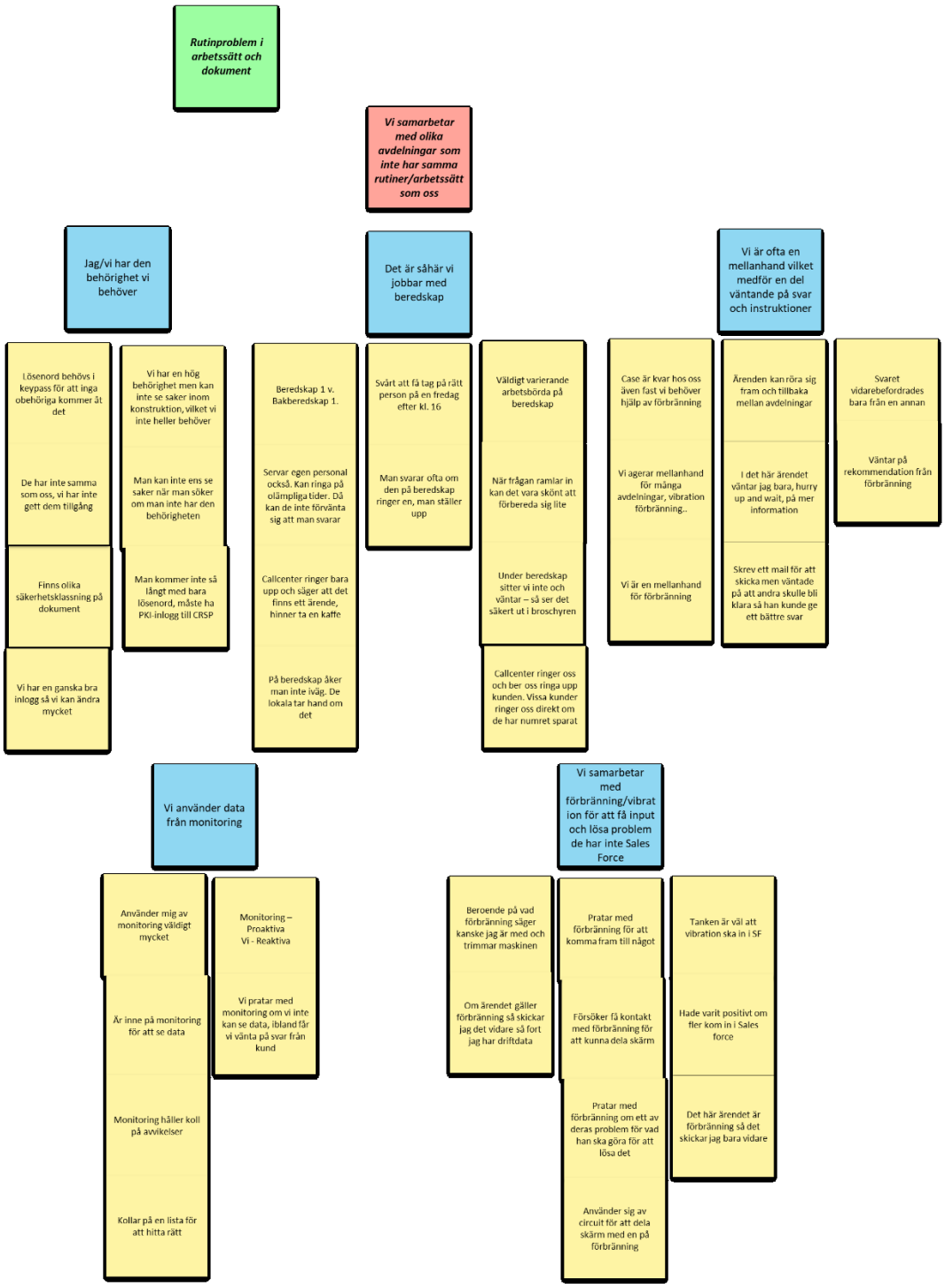


Bild 3. Del av affinitetsdiagrammet som tar upp bland annat att de tar hjälp av och sammanbetar med andra avdelningar.

Nästkommande tre bilder, se bild 4, 5 och 6, visar en del av affinitetsdiagrammet som tar upp problematiken med att informationen är utspridd och att de sökfunktioner som finns idag är bristfälliga. De anser att det finns för många program och skulle vilja ha ett enda system som de använder sig av som hanterar samtliga system de behöver tillgång till. Ytterligare

problematik kring sökning av information är att flera system de använder sig av har svåra sökfunktioner, vilket bidrar till att supportteknikerna upplever att systemen är dåliga.

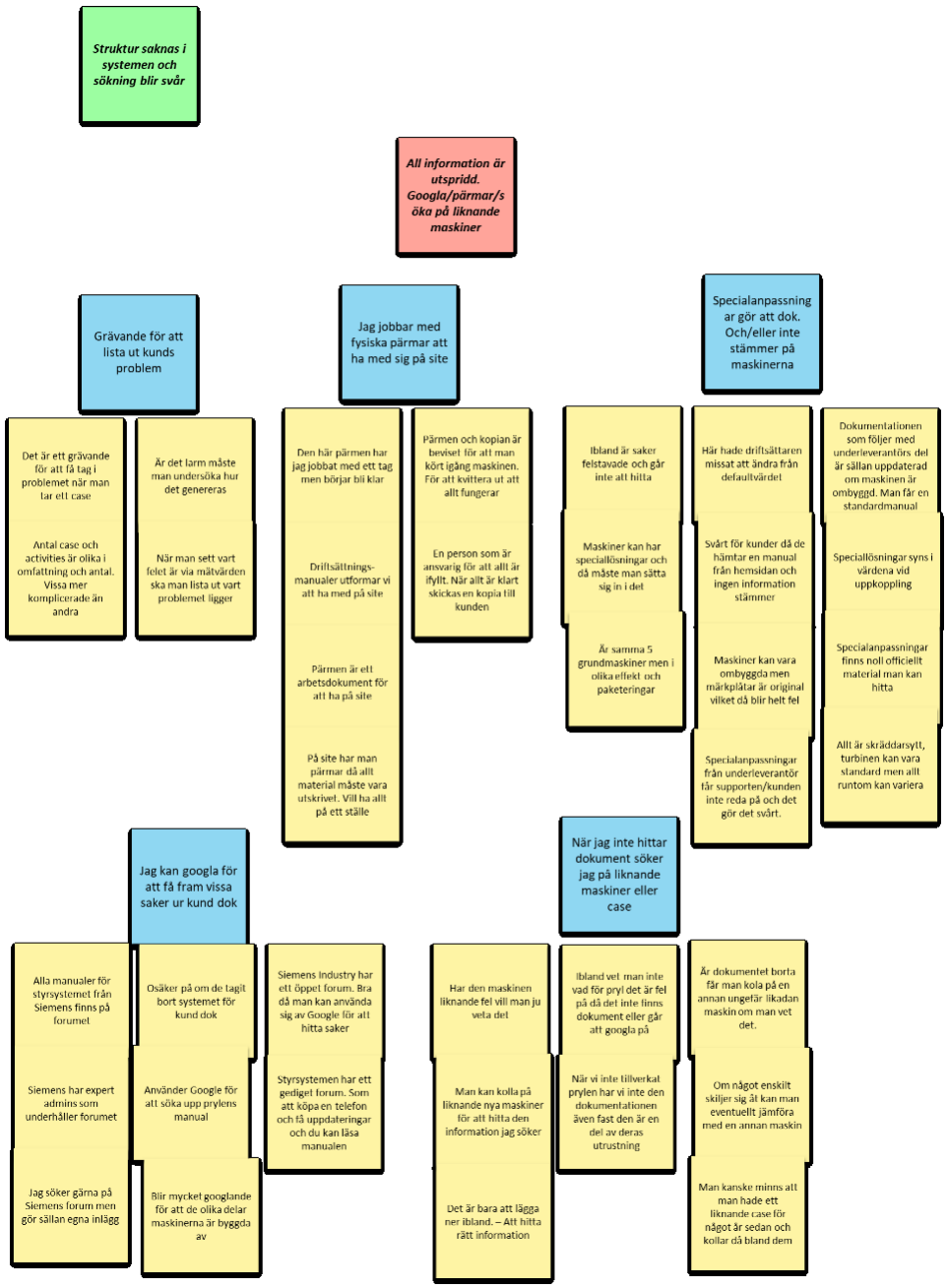


Bild 4. Del av affinitetsdiagrammet som beskriver att information är utspridd både digitalt och fysiskt.

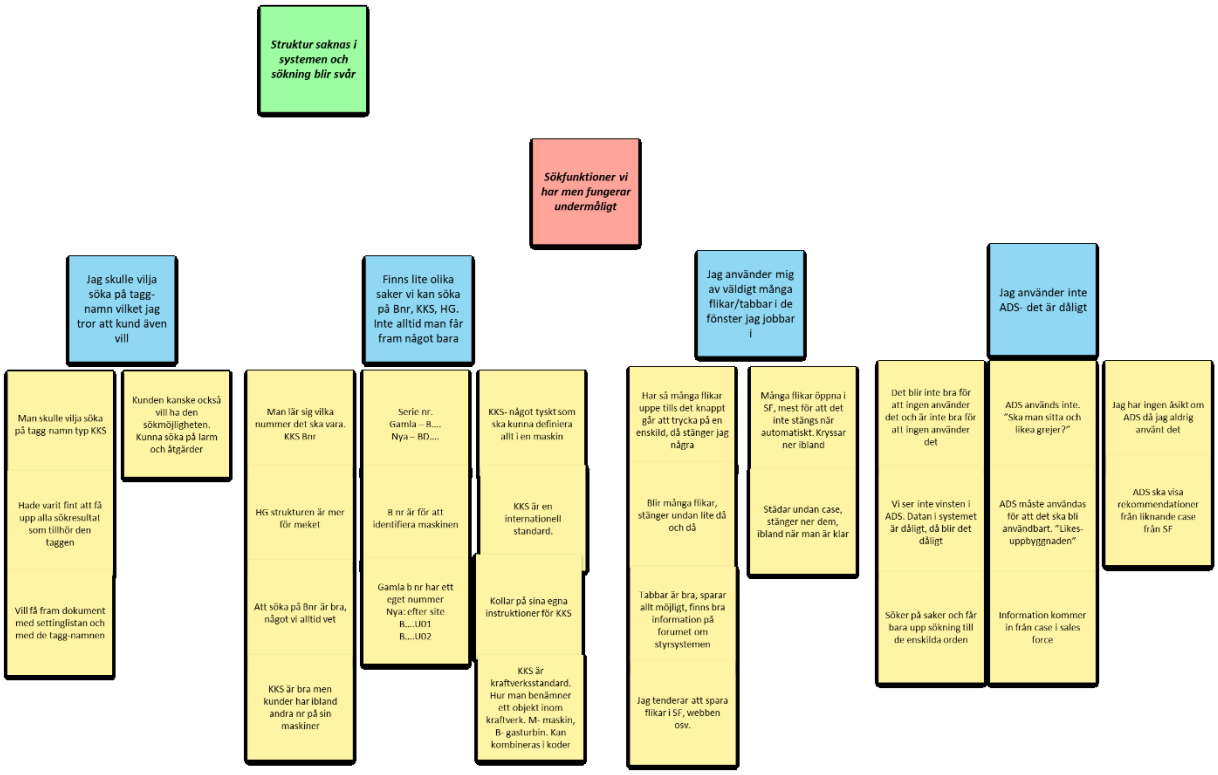


Bild 5. Del av affinitetsdiagrammet som visar på hur de skulle vilja att det var att söka information och hur det är idag.

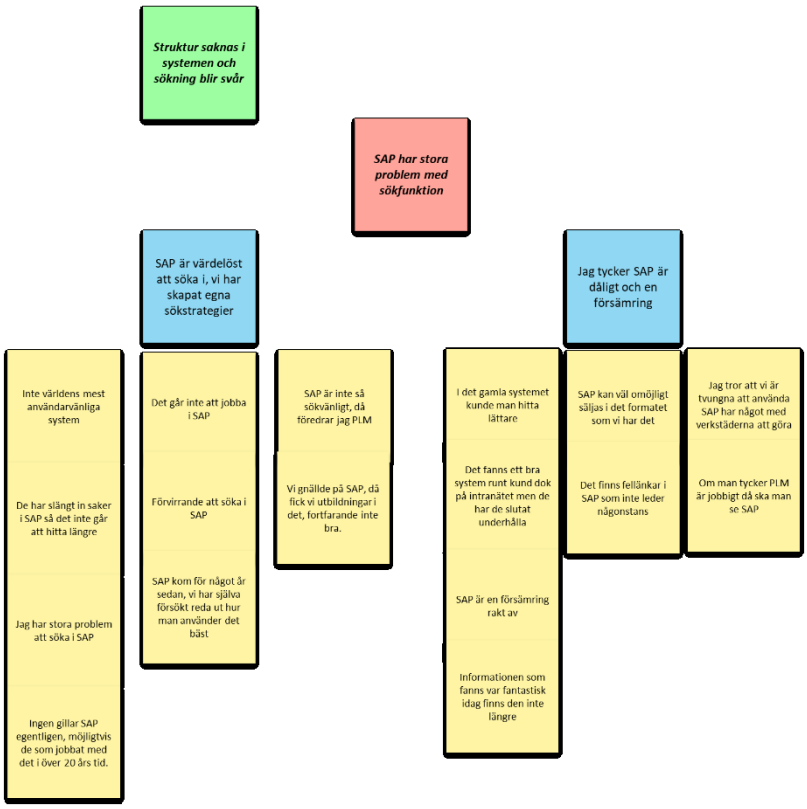


Bild 6. Del av affinitetsdiagrammet som tar upp problematik i system de har kring att hitta information.

Slutligen visar den sista bilden (se bild 7) problematiken som supportteknikerna upplever kring dokumenthantering. Dokumenten som skapas kan döpas på olika sätt, det finns tvivelaktiga rutiner för att införa dokument i systemen de har och saknas dokument så skapas det inga nya.

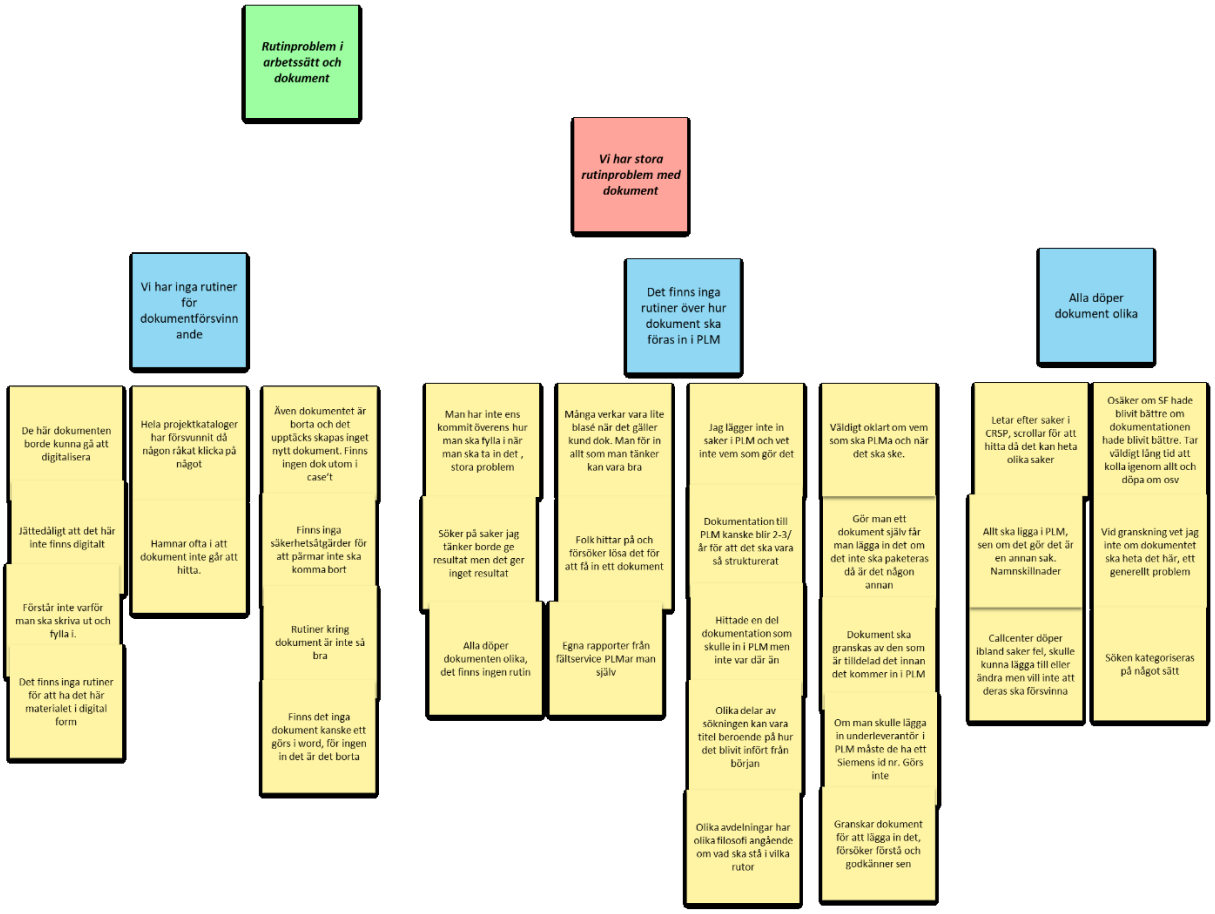


Bild 7. Del av affinitetsdiagrammet som beskriver den upplevda bristen i dokumenthanteringen.

4.2 Kravformulering och prioritering

Här beskrivs den persona och det kontextscenario som skapades för att ta fram krav. Därefter presenteras kraven och prioriteringen av dessa.

4.2.1 Persona och kontextscenario

Den persona som skapades sammanfattar de individer som intervjuades under datainsamlingen. Det slutgiltiga resultatet kan ses på bilden nedan (se bild 8). Personan är fiktivt och enbart skapad utifrån tolkningar av den data som samlats in. Bilden som används är hämtad från Google och har ingen koppling till de som arbetar på Siemens mer än att den ska efterlikna den person som beskrivs i personan.

Bild 8. Persona av en fiktiv person vid namn Karl Axelsson.

Med denna persona som grund och utifrån affinitetsdiagrammet skapades detta kontextscenario:

> Karl får in ett case (fall) där en kund behöver hjälp med ett problem med sin maskin. För att lösa detta problem vet Karl att han behöver hitta specifik information som ligger i ett dokument under ett visst taggnamn. Därför klickar sig Karl direkt in till chatboten för att be den att hitta denna information. Hade Karl vetat vad för dokumentnummer dokumentet har hade han kunnat skriva detta till chatboten och den hade då gett det specifika dokumentet. Karl vet dock i detta fall inte dokumentnumret och skriver därför att han söker efter en viss typ av information, med hjälp av taggnamnet, till chatboten. Chatboten förslår information som Karl kan tänkas vara ute efter. Därefter kan Karl snabbt hitta informationen och ge svar till kunden om vad som ska göras.

Det skapades enbart ett kontextscenario då det var denna typ av uppgift som det upplevdes vara mest fokus på från supportteknikernas synvinkel. Utifrån den datainsamling som genomförts har det visat sig finnas problem i grundläggande funktioner i systemen de använder idag. Däribland var de problem som uppdagades att det bland annat var för många system att leta efter information i samt att sökfunktionerna är bristfälliga. Utifrån dessa problem var det denna typ av uppgift som ansågs vara möjlig att implementera i en chatbot. Kontextscenariot skulle kunna delas upp i separata scenarion, dock ansågs dessa scenarion vara för lika för att skapa någon mening i att dela upp dessa. Fokuset kring dokumentnummer och taggnamn kommer från

affinitetsdiagrammet där det tydliggjordes att det var det de använde sig av mest för att få fram information samt att det var något de önskade att få använda sig av.

4.2.2 Sammanställda och prioriterade krav

Utifrån den persona och det kontextscenario som skapades, samt utifrån bakgrundslitteratur och affinitetsdiagrammet kunde datakrav, funktionskrav, kvalitetskrav och begränsningar tas fram. Inom parentes beskrivs det var kraven extraherats ifrån för att skapa en spårbarhet som tidigare nämnts är bra att ha. Datakraven behöver specificeras mer vid framtida implementering, men för detta projekt ansågs detta inte nödvändigt då denna del inte skulle behandlas. Nedan beskrivs de framtagna kraven.

Datakrav:

- Dokument med textuell och eventuell visuell information om den maskin eller maskindel som är relevant för kundproblemet (från kontextscenario)
- Dokument har dokumentnummer och/eller taggnamn (från kontextscenario)

Funktionskrav:

- Kunna få chatboten att hitta dokument med specifikt dokumentnummer i alla nuvarande system (från kontextscenario och persona)
- Kunna få chatboten att hitta dokument eller information med specifikt taggnamn i alla nuvarande system (från kontextscenario och persona)
- Kan visa var mer information om själva chatboten kan hittas (från bakgrundslitteraturen)
- Möjligt att lämna feedback kring hur chatboten svarar och agerar (från bakgrundslitteraturen)

Kvalitetskrav:

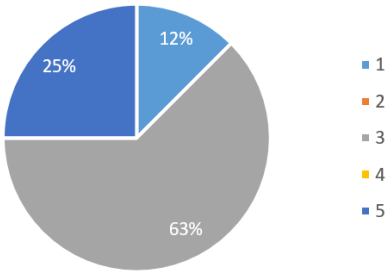
- Snabb tillgång till den sökta informationen (från persona)
- Lättillgänglighet till chatboten (från kontextscenario)
- Lätt att hitta den information man söker (från persona)
- Chatboten har sökmöjligheter som är enkla att förstå (från persona)
- Tydligt vad chatboten kan och inte kan göra för något (från bakgrundslitteratur)
- Tydligt hur chatboten ska användas och hur den fungerar (från bakgrundslitteratur)

Begränsningar:

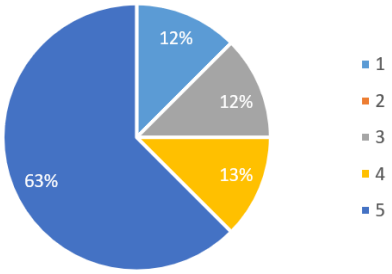
- Ska kunna hantera att dokumentation är sekretessbelagd och ibland inte tillgänglig (från affinitetsdiagrammet, se bild 3)
- Ska ha flexibilitet för att hantera varierande rutiner för dokumenthantering och dokumentnamngivning (från affinitetsdiagrammet, se bild 7)
- Ska kunna hantera att allt material inte är digitalt (från affinitetsdiagrammet, se bild 4)
- Körs på en plattform som supportteknikerna använder sig av (från bakgrundslitteraturen)

Därefter prioriterades funktions- och kvalitetskraven då dessa har ett användarfokus och är de krav som påverkar designen av chatboten. Övriga krav fick samma kravprioritet. Anledningen till att enbart funktions- och kvalitetskraven prioriterades var för att dessa var de enda som var av relevans för supportteknikerna som kunde ge en påverkan för detta projekt. Övriga krav påverkar framtida implementeringsprocesser. Utifrån de 11 supportteknikerna som kontaktades medverkade totalt 8 i denna prioritering.

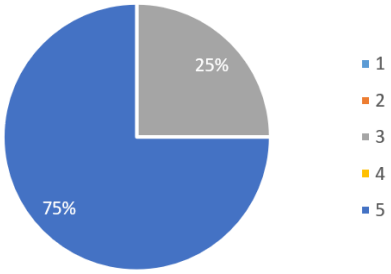
Lättillgänglighet till chatbot



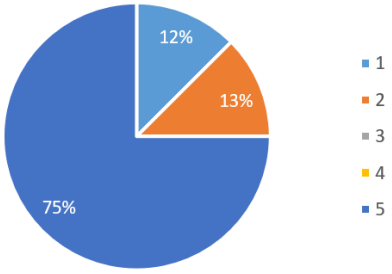
Kunna få chatboten att hitta dokument med specifikt dokumentnummer i alla nuvarande system



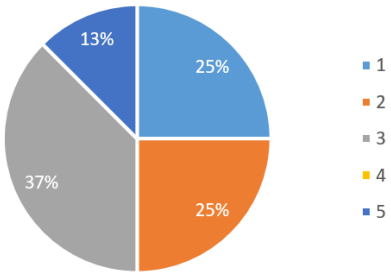
Snabb tillgång till den sökta informationen



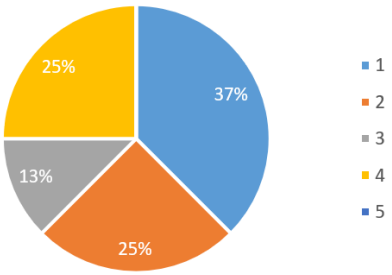
Kunna få chatboten att hitta dokument eller information med specifikt taggnamn i alla nuvarande system



Kan visa var mer information om själva chatboten kan hittas



Möjlighet att lämna feedback kring hur chatboten svarar och agerar



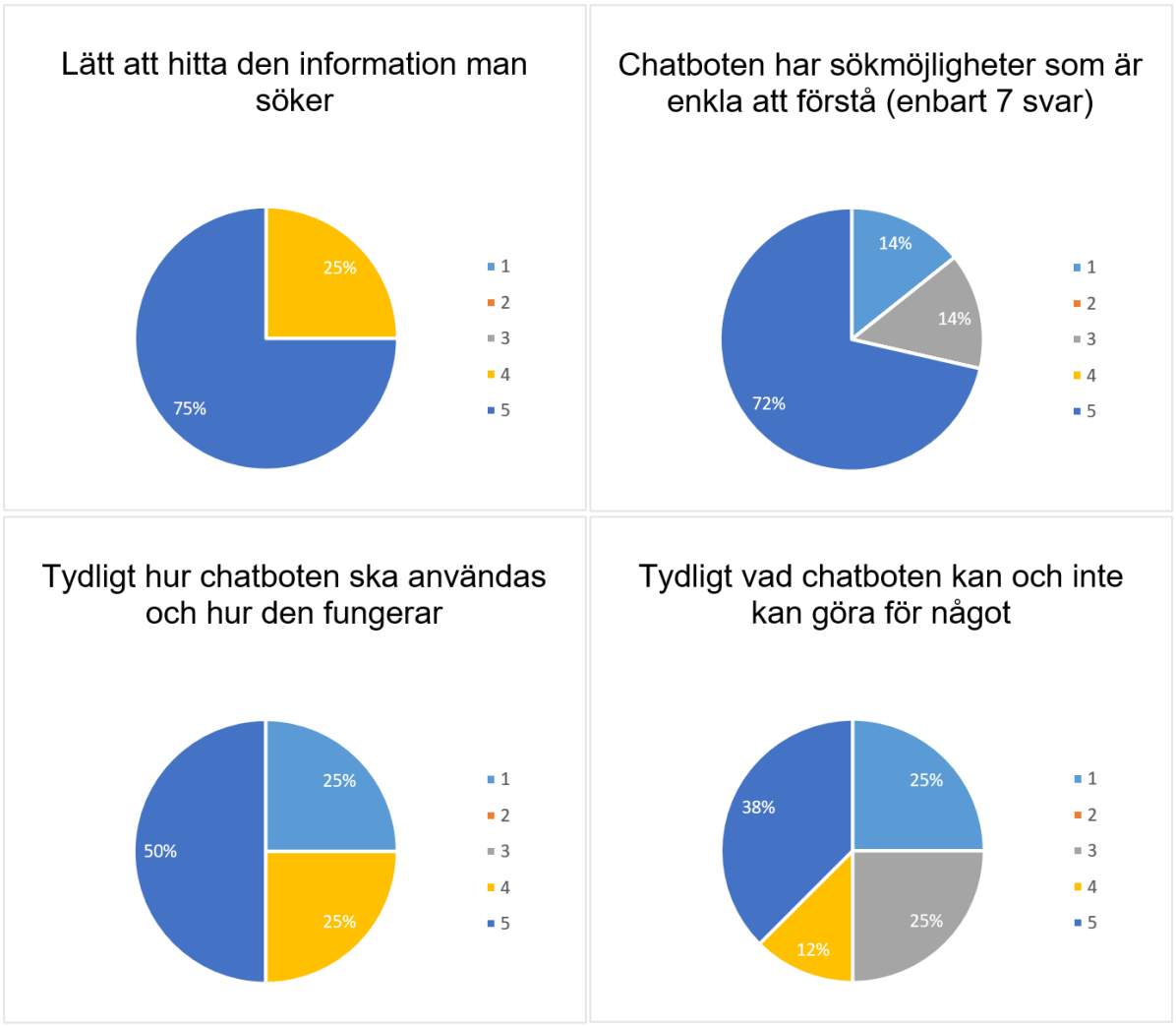


Bild 9. Svaren från formuläret där supportteknikerna skulle poängsätta kraven mellan 1–5.

Vid undersökning av de enskilda svaren var den en deltagare som svarade att alla krav som involverade en chatbot inte var av så stor vikt, se tabell 1. Detta bidrog till att de flesta krav alltid hade en etta i sina poängsättningar.

Tabell 1. Visar en översikt över hur de enskilda personerna svarade

Deltagare	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*Medel (Prio)*
Lättillgänglighet till chatboten	3	3	3	1	5	5	3	3	3,25 (8)
Kunna få chatboten att hitta dokument med specifikt dokumentnummer i alla nuvarande system	5	5	5	1	5	3	5	4	4,125 (5)
Snabb tillgång till den sökta informationen	3	5	5	5	5	5	5	3	4,5 (2)
Kunna få chatboten att hitta dokument eller information med specifikt taggnamn i alla nuvarande system	5	5	5	1	5	5	2	5	4,125 (4)
Kan visa var mer information om själva chatboten kan hittas	5	3	2	1	1	3	2	3	2,5 (9)
Möjlighet att lämna feedback kring hur chatboten svarar och agerar	1	4	3	1	1	2	2	4	2,25 (10)
Lätt att hitta den information man söker	5	5	4	5	5	5	5	4	4,75 (1)
Chatboten har sökmöjligheter som är enkla att förstå	-	5	5	1	5	5	3	5	4,14 (3)
Tydligt hur chatboten ska användas och hur den fungerar	1	5	4	1	5	5	5	4	3,75 (6)
Tydligt vad chatboten kan och inte kan göra för något	1	5	3	1	3	4	5	5	3,375 (7)

Då en supporttekniker missade att poängsätta ett krav räknades ett medelvärde ut på poängen för varje krav (se bild 9). Prioriteringen av dessa krav blev därmed som följande:

1. Lätt att hitta den information man söker
2. Snabb tillgång till den sökta informationen
3. Chatboten har sökmöjligheter som är enkla att förstå
4. Kunna få chatboten att hitta dokument eller information med specifikt taggnamn i alla nuvarande system
5. Kunna få chatboten att hitta dokument med specifikt dokumentnummer i alla nuvarande system
6. Tydligt hur chatboten ska användas och hur den fungerar
7. Tydligt vad chatboten kan och inte kan göra för något
8. Lättillgänglighet till chatboten
9. Kan visa var mer information om själva chatboten kan hittas
10. Möjlighet att lämna feedback kring hur chatboten svarar och agerar

4.3 Slutgiltig prototyp

Prototypen av chatboten gjordes i storlek av en Iphone 6/7 då detta ansågs vara mest likt storleken av en chatbot. När en konversation med en chatbot inleds skickar chatboten först en fråga till användaren, eller i detta fall en supporttekniker, se bild 10.

Bild 10. Start av interaktion med chatbot.

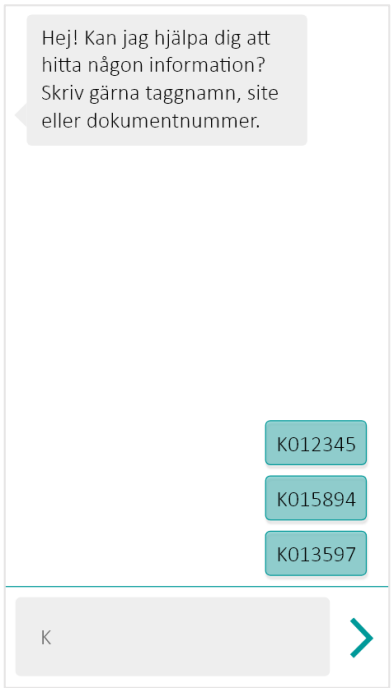


Bild 11. Förslag på taggnamn utifrån vad användaren skriver.

När supportteknikern sedan börjar skriva in ett taggnamn dyker förslag upp (se bild 11). På samma sätt är det tänkt att fungera om användaren börjar skriva in ett namn på en site (vilket är platsen den eftersökta maskinen finns på). Om mer av taggnamnet, eller sitenamnet, skrivs kommer förslagen att ändras samt vara allt mer lika det taggnamn eller den site som skrivits. Dessutom kommer antalet förslag bli färre, exempelvis när det inte finns så många taggnamn med den kombinationen som skrivits så kanske bara ett förslag visas. Finns det inga förslag så finns det ingen dokumentation med det taggnamnet eller för den siten. De förslag som finns går att trycka på och leder till att det valda förslaget blir supportteknikerns svar till chatboten (se bild 12). Med hjälp av förslagen som dyker upp när användaren börjar skriva ett svar underlättar det för användaren bland annat genom att de slipper skriva in hela svaren själva. Vilket bidrar till uppfyllande av kravet att det skulle vara (6) tydligt hur chatboten ska användas och hur den fungerar. Detta leder i sin tur till att kravet kring att det ska vara (7) tydligt vad chatboten kan och inte kan göra, eftersom chatboten begränsar användaren till att svara på specifika frågor. Dessutom får de en direkt återkoppling till om den informationen de söker finns eller inte. Detta tillsammans med att chatboten ber användaren skriva site, taggnamn eller dokumentnummer implementerades för att uppnå två av kraven. Dessa krav var att det skulle vara (1) lätt att hitta informationen som eftersökes samt att (3) chatboten skulle ha sökmöjligheter som var enkla att förstå. Genom att beskriva för användaren vad chatboten vill veta för att kunna söka efter specifik information blir det tydligt och enkelt för användaren att veta vad som ska göras. Det minskar risken för att användare inte ska förstå hur de ska svara chatboten. Detta bidrar även till att kraven kring att chatboten ska (4, 5) kunna hitta dokument och information med specifika taggnamn eller dokumentnummer uppfylls.

Därefter svarar chatboten med att be användaren, eller supportteknikern, att ge ytterligare svar angående vad för information som eftersökes. När supportteknikern gett sitt svar, ber chatboten att användaren ska vänta tills att den funnit den specifika informationen eller relaterad information (se bild 13). Om användaren skriver site istället för taggnamn får personen upp förslag kring eventuella taggnamn som det skulle kunna gälla. Därefter är den inne på samma spår som visas av prototypen. Detta har inte designats då det utgått från att användarna kan söka på taggnamn då det är något som förväntas att de kan idag samt att det är något de efterfrågat att kunna söka på. Dock gavs site som förslag att skriva för att visa vad för information som kan efterfrågas om de exempelvis inte skulle minnas eventuella taggnamn. Hade användaren istället skrivit in dokumentnummer hade det sett ut på liknande sätt, men utan att chatboten ber om ytterligare information då det är ett specifikt dokument som efterfrågas.

Denna typ av sökning ansågs vara mest relevant då de dels får möjlighet att söka på det sätt de önskat, alltså med taggnamn och eventuell specifik information som de söker efter (se bild 5). Att ge förslag efterliknar också hur sökmotorer fungerar idag, exempelvis Google, vilket gör att det finns en vana och eventuell bekvämlighet i att få förslag och slippa skriva ut allt själv.

Design på skrivrutan begränsades till att efterlikna hur de ser ut på många chattplattformar och för att tydligt avgränsa det användaren skriver från själva chatten. Knappen applicerades för att tydliggöra för användaren hur man gör för att skicka meddelanden. Många chatplattformar använder sig av returknappar eller pilar av olika slag. Därför placerades denna pil in för att den inte skulle ta upp så mycket plats, men även för att göra det lätt för användaren att skicka iväg ett meddelande.

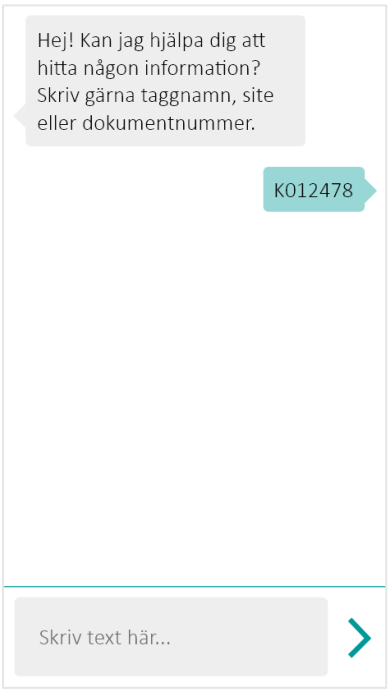


Bild 12. Förslag blir till svar till chatboten.

Bild 13. Användaren har gett ytterligare information och ombes vänta.

När chatboten hittat relevant dokumentation skickas detta tillbaka till supportteknikern så att hen kan öppna dokumentet och hitta informationen. De två högst prioriterade kraven, (1) lätt att hitta den information man söker och (2) snabb tillgång till den sökta informationen, var de som ledde till designbesluten om att bland annat chatboten skulle förse med en direktlänk till dokumentet. Direktlänken gör det snabbt för användaren, eller supportteknikern i detta fall, att komma åt den sökta informationen. Detta beslutades vara den bästa lösningen eftersom en chatbot ska göra att uppgiften löses snabbare och enklare än om chatboten inte används.

Direktlänken till dokumentet ges tillsammans med en fråga där chatboten undrar ifall informationen var den som eftersöktes (se bild 14). Under frågan som chatboten ställer finns en

länk som är tänkt att ta supportteknikern till dokumentet och ger då möjlighet till supportteknikern att direkt undersöka ifall det är korrekt dokument. Ovanför textrutan som supportteknikern kan skriva i finns två alternativ, ja och nej. Dessa är för att besvara frågan chatboten ställt. Svarar supportteknikern ja ser det ut som i bild 15, om svaret är nej ser det ut som i bild 16. Trots att kravet kring att det ska vara (10) möjligt att lämna feedback kring hur chatboten svarar och agerar var lägst prioriterat, valdes detta att implementeras. Detta för att det kunde implementeras på ett naturligt sätt. Dessutom är det bra för att chatboten ska kunna lära sig vilka dokument som är relevanta för specifika sökord och för att förbättra användarupplevelsen. För att öka chansen att feedback ges valdes det att implementera ett simpelt gränssnitt i form av ja- och nejknappar. Dessa ska besvara frågan om den länk de fått var det de sökte eller inte. Med knappar slipper de skriva svaren själva och kan med ett enkelt knapptryck besvara på frågan. Dessutom uppmuntrar det dem till att ge feedback istället för att ignorera frågan, som de i många andra fall troligtvis hade gjort.

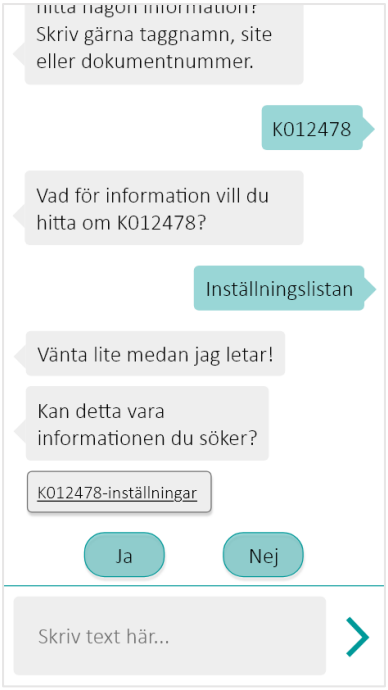


Bild 14. Länk förses och ger användaren.

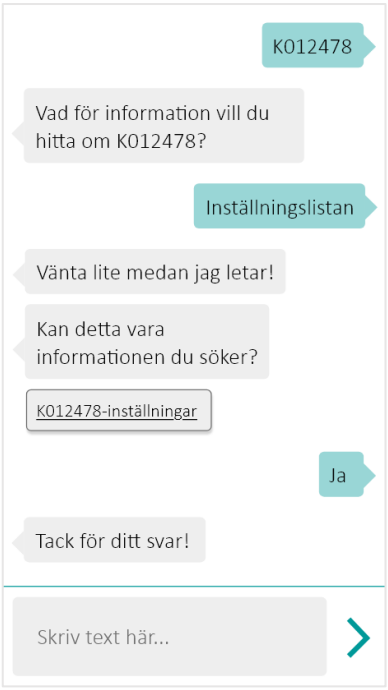


Bild 15. När användaren svarar ja möjlighet att bidra med feedback.

Det ansågs även viktigt att chatboten ger feedback till användaren av vad som förväntas av hen och vad som kan förväntas av chatboten. Genom att skriva meddelandet ”Vänta lite medan jag letar!” minskar risken för eventuell frustration från användaren ifall att det skulle ta lite längre tid än önskat att hitta relevant information. På så vis får de tydliga instruktioner om att chatboten arbetar med det och återkommer när informationen hittats. Dessutom är det viktigt att berätta för användaren vad som förväntas som svar ifall att den informationen som föreslogs inte var

den rätta. Vilket leder tillbaka till att uppfylla kraven kring bland annat att (3) chatboten ska ha sökfunktioner som är enkla att förstå.

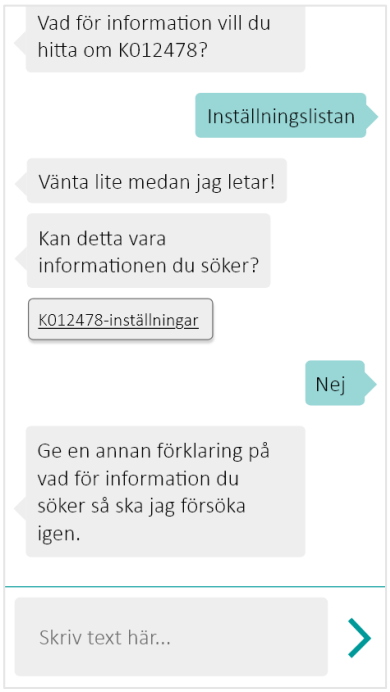


Bild 16. När användaren svarar nej.

För att se alla bilder över den fullständiga prototypen se Appendix A – Fullständig prototyp, steg för steg. Där beskrivs och presenteras det exakt hur prototypen ser ut för varje steg.

5. Diskussion

I denna sektion kommer inledningsvis resultatet av detta projekt att diskuteras, där stora delar av diskussionen ligger kring just prototypen som skapades. Därefter kommer det att diskuteras kring möjligheten att implementera en chatbot i Siemens system.

5.1 Affinitetsdiagram, persona och krav

Som nämndes i resultatet var det enbart delar av affinitetsdiagrammet som redovisades i resultatet. Anledningen till detta var för att det enbart var dessa typer av delar från affinitetsdiagrammet som var av relevans för detta projekt och som påverkade resultatet. Sedan skapades en persona utifrån dessa delar samt den subjektiva bilden som formades av de som blev intervjuade. Som nämndes i metodbeskrivningen gjordes ingen skillnad på individerna som intervjuades. För eventuell framtida forskning och skapande av personas skulle det vara bra att göra en undersökning på individnivå för att se ifall de kan delas upp i flera personas. Dessutom skulle det vara bra att undersöka ytterligare kring personliga egenskaper för att vara säkert på att den bild som skapades i detta projekt är korrekt eller ifall det möjligtvis behöver korrigeras. Detta är viktigt för den eventuella kommande implementeringen då det är viktigt att chatboten är anpassad för så många som möjligt som är tänkta att använda den. Eventuella skillnader hos användarna skulle kunna vara viljan i att lära sig ett nytt program eller system, teknikintresset, preferenser i om chatboten ska vara personlig eller inte, exakt hur de skulle vilja kunna söka på saker och så vidare. Med dessa skillnader kan det komma krav på hur en chatbot ska fungera och se ut. Då alla kan ha olika åsikter kring detta är det viktigt att de, som i detta projekt, prioriteras.

Utifrån de krav som framställdes är det ett par som inte kunde uppfyllas i just detta projekt, dock är de viktiga att ta med för framtida implementering. Just datakraven och begränsningarna är sådana krav som har en stor påverkan på hur den ska/bör implementeras och utvecklas samt vad man behöver fundera mer kring innan en implementering genomförs. Exempelvis måste man vid utveckling ta stor hänsyn till den sekretess som ligger kring dokumenten, detta kräver med stor sannolikhet mycket arbete för att fel information inte ska hamna i fel händer. Dessa krav måste för framtida implementering specificeras ytterligare och utökas. Då detta enbart var en inledande studie finns det fler krav som kommer att upptäckas ju längre in i processen man kommer. Som beskrevs i metodavsnittet finns det ett inneboende beroende mellan problemformulering och lösningsförslag (Arvola, 2019). Ju större förståelsen för problembilden

är desto fler krav kan framställas. Som förslaget ovan, att öka sin förståelse för exakt vad det är användarna söker för funktionalitet, bidrar till fler krav som beskriver i mer detalj kring hur chatboten ska se ut och fungera. Ökas alltså förståelsen för vad som efterfrågas och även vad en eventuell lösning skulle kunna vara kan fler krav dyka upp. Inledningsvis är det som Arvola (2019) nämner inte det viktigaste, om ens möjligt, att få med alla krav. Detta är något som byggs på under tiden som ytterligare undersökningar och iterationer genomförs.

Datakraven och begränsningarna användes, som tidigare nämnts, inte vid prioritering av kraven. Detta bland annat på grund av det som beskrivits ovan, men även att de inte var av relevans för supportteknikerna. Dessa är inte relevanta för supportteknikerna då detta påverkar bakomliggande kod i hur chatboten fungerar och är därmed inget som supportteknikerna ser. Det finns en stor risk att de hade varit svåra att prioritera för dem då de skulle kunna anses vara lika viktiga vid implementering. För denna uppsats låg även fokuset mer på själva designen av chatboten, och inte den bakomliggande tekniken, därmed prioriterades inte dessa krav. På grund av det som beskrevs ovan kan det dock bli relevant i framtiden, och då är det viktigt att tänka på vilka personer som bör prioritera dessa krav. Datakrav och begränsningar bör eventuellt prioriteras av personer som har högre kunskap kring de system som finns idag och även en del kunskap inom chatbotteknik.

Prioriteringarna som genomfördes resulterade i liknande svar från samtliga supporttekniker. För de flesta krav hade de ganska lika åsikter men de kunde ibland skilja sig åt. Bland annat var det som nämndes i resultaten en deltagare som svarade att det var lägst prioritet på alla krav som involverade funktionaliteter och kvalitéer av en chatbot. Denna person kan ha ansett att just en chatbot inte var särskilt relevant. Dock ansåg denna person att det var av hög relevans att de snabbt får tillgång till den sökta informationen samt att det ska vara lätt att hitta den informationen man söker. Detta innebär med stor sannolikhet att det för personen inte är viktigt att det är en chatbot utan att det åtminstone förs in någon typ av teknik som gör det lättare och snabbare att hitta samt få tag på information. I övrig var det som sagt väldigt lika åsikter kring kraven där de kunde skiljas åt till viss del i vissa krav. Anledningen till att det skiljer sig är svårt att tolka mer än att några krav är viktigare för vissa än för andra. Därmed skulle även detta vara ett område att undersöka vidare för att se varför åsikterna skiljer sig åt för att få en bättre bild av vad det är de söker i ett sådant här typ av system. Dessutom hade det varit bra för att få en tydligare förståelse till varför de prioriterade som de gjort. För att få denna förståelse hade man om möjligt kunnat utföra denna prioritering tillsammans med varje person för att under tiden de prioriterar kunnat få med åsikter om varför de prioriterade som de gjorde. På så vis skapas

en tydligare bild kring deras tankegång och varför de prioriterar som de gör. Vid sådana fall är det dock väsentligt att inte prata för mycket så att det påverkar deras åsikter och tankar. Ett annat sätt hade varit att ge dem möjlighet att skriva en kort text om hur de resonerade kring deras poängsättning eller prioritering om ett formulär används, som i detta projekt. Dock är det svårt att få med alla åsikter och tankar i textformat, samt att det i detta fall skulle kunna leda till att de väljer att inte göra prioriteringen eller inte svarar kring hur de resonerade. Bäst resultat hade troligtvis varit om en diskussion hade förts tillsammans med varje deltagare, antingen under tiden som prioriteringen sker eller även kort tid efter. Sker den efter måste detta ske tätt inpå för att med större säkerhet få svar på hur de tänkte kring deras poängsättning eller prioritering.

I ett av kraven var det en som antingen missat att svara, eller som aktivt valde att inte svara, detta hade kunnat undvikas genom att göra det obligatoriskt att poängsätta alla krav. Detta hade lett till att om de exempelvis missat att poängsätta ett krav så hade de fått en notis om det, samt att de inte hade kunnat lämna in sin poängsättning utan att ändra på det. Hade denna poängsättning inte missats hade det kunnat leda till en eventuell förändring i prioritering av krav. Dock borde det inte orsakat någon större förändring då medelvärdena för poängsättning användes. En annan eventuell problematik med denna prioriteringsmetod är att deltagarna kan poängsätta genom att enbart ge högsta eller lägsta poäng, vilket gör det svårt att sammanställa och prioritera kraven. Detta var dock inte fallet för just detta projekt då de svarade tillräckligt olika för att göra det möjligt att skapa en prioriterad lista av kraven. Det var två av deltagarna som helt eller i princip enbart gav högsta eller lägsta poäng på varje krav. För att förhindra detta problem för fortsatta undersökning är det av denna anledningen bättre att använda sig av att de får prioritera alla krav genom att sätta varje krav i en hierarkisk ordning. Dock var poängsättningen en bra metod att använda sig av för detta projekt för att öka chanserna att få så många svar som möjligt. Då det enbart var två som svarade på detta sätt var det som sagt fortfarande möjligt att skapa en prioriterad lista. På grund av det här är det viktigt att tänka på vad för metod som är bäst för just den undersökning som ska genomföras, är det svårt att få svar kan det vara bra med poängsättningsmetoden. I annat fall kan det eventuellt vara bättre att få dem direkt prioriterade av deltagarna.

5.2 Prototyp

Under denna del av diskussionen kommer prototypen att diskuteras, först och främst utifrån de krav som tagits fram för detta projekt för att därefter gå in på vad för möjligheter denna prototyp har.

5.2.1 Prototyp i förhållande till prioriterade krav

Prototypen som skapades i detta projekt grundades som tidigare nämnts i de prioriterade kraven. Direktlänken designades med stor grund i kraven att det skulle vara lätt att hitta och få snabb tillgång till den sökta informationen. Det skulle även vara en möjlighet att chatboten ger förslag till användaren var informationen kan finnas någonstans och hur de ska göra och söka för att komma dit. Detta skulle eventuellt vara en funktionalitet som passar in bra i hur systemen och programmen fungerar som Siemens har idag. All sekretessbelagd information där olika typer av inloggning krävs gör att det kan vara svårt att implementera en chatbot som ger direkt tillgång till sådana dokument. Skulle chatboten enbart förse användaren med instruktioner om var informationen finns och hur supportteknikerna kan söka efter informationen så finns det dock en stor risk att de inte skulle använda chatboten. Utifrån uppfattningen av vad supportteknikerna vill ha så skulle en sådan lösning som enbart ger instruktioner göra att de hellre sökte efter informationen själv än att använda och få hjälp av chatboten. Därmed ansågs lösningen med en direktåtkomst vara bättre då det ansågs vara mer troligt att supportteknikerna skulle använda chatboten. Som tidigare nämnts, om det är lättare eller eventuellt ingen skillnad i att genomföra uppgiften utan chatboten än med chatboten är det större risk att den inte används.

Kravet om att göra det lätt att hitta information som eftersökes ansågs gå ihop en del med kravet om att chatboten ska ha sökmöjligheter som är enkla att förstå. Detta implementerades i designen av chatboten och prototypen på så vis att chatboten berättar för användaren vad som händer och vad användaren bör svara. Exempelvis skriver chatboten att användaren först ska skriva namn på site eller taggnamn för att därefter skriva ytterligare information om exempelvis det taggnamn som användaren skrivit. Detta minskar som tidigare nämnts risken för att användare inte ska förstå hur de ska svara chatboten. Vilket i sin tur bidrar till att kraven kring att chatboten ska kunna hitta dokument och information med specifika taggnamn eller dokumentnummer uppfylls. Förslagen underlättar för användaren att slippa skriva in hela svaren själv, samt att personen får direkt återkoppling om informationen som eftersökes finns eller inte.

Feedbackmöjligheten ansågs, som tidigare nämnts, vara bra att implementera för att visa hur det kan göras på ett simpelt sätt för att kunna förbättra användarupplevelsen av chatboten och chatbotens funktionalitet. Supportteknikerna anser eventuellt att det är överflödigt och slöseri med tid att behöva lämna feedback. För att öka chanserna för detta skapades ja- och nejknappar, då det är snabbare och smidigare att bara behöva göra ett knapptryck. Detta uppmuntrar även till att ge feedback då de direkt får möjlighet att trycka på knappar istället för att skriva ett svar själva. Hade de behövt skriva själva finns det en stor risk att de inte svarat, detta på grund av den låga prioriteten på kravet om möjligheten att ge feedback.

Det var två prioriterade krav som valdes att inte implementeras i design av chatboten, dessa var (8) lättillgänglighet till chatboten samt att den (9) kan visa var mer information om själva chatboten kan hittas. Detta eftersom de är och skulle kunna vara beroende av hur det ser ut kring chatbotens chatfönster. Lättillgängligheten är direktkopplat till kontexten vilket bidrog till att detta inte valdes att implementeras i designen. Detta är något som måste undersökas ytterligare för att veta var och hur en chatbot ska implementeras i Siemens system då det är beroende av många olika faktorer. I vilket system bör detta implementeras eller ska det vara en fristående funktion? På vilket sätt ska chatboten ha tillgång till de olika systemen och hur ska den få tillgång till de olika systemen? Att i sin tur visa var mer information om själva chatboten kan hittas kan på samma sätt ligga i anslutning till chatbotens chatfönster eller intill eventuell chatbotikon. Detta kan till stor del också bero på kontexten. En lösning skulle kunna vara att chatbotfönstret har en header med eventuell profilbild på chatboten för att skapa en viss personlighet till den. Dessutom skulle det kunna finnas någon typ av knapp i headern som ger tillgång till ytterligare information om chatboten. Dock var detta enbart en första prototyp. Det kommer att behöva skapas fler designförslag och prototyper samt iterationer med dessa för att skapa en chatbot som uppfyller alla krav, är användarvänlig och anpassad till Siemens grafiska profil. För denna prototyp lades inte så mycket fokus på det grafiska utan mest fokus låg på användarvänlighet, där färgerna som användes var tagna från Siemens logotyp.

5.2.2 Prototypens möjligheter

Prototypen som skapades för detta projekt visar inte alla möjliga utvägar för hur designen skulle kunna se ut. Detta på grund av tidsbegränsningen projektet har haft, samt att det enbart var en första prototyp som inte genomgått några iterationer. Bland annat implementerades inga profilbilder eller liknande då detta inte finns i de systemen som Siemens använder sig av idag och att enbart ge chatboten en profilbild ansågs överflödigt för denna prototyp. I framtida

designförslag skulle detta kunna ses över och eventuellt implementeras för att göra chatboten mer personlig.

Smådetaljer i designen av chatboten valdes det att inte ha något större fokus på då det viktigaste med denna prototyp var att få fram användarvänligheten. Detaljer är viktiga för fortsatta arbetet med chatbotutvecklingen för att förbättra helhetsupplevelsen av chatboten. Förslag på detaljer skulle kunna vara tidstämplar när meddelanden skickats, profilbilder, hur stora chatrutorna är, färgval och så vidare. Detta gå mer in under grafisk design som kan vara bra att tänka på och ta del av i fortsatta arbetet. Designandet av en chatbot blir ganska begränsad på så vis att det är någon typ av konversation som ska ske där både chatboten och användaren ska förse med någon fråga, kommentar eller svar. Då chatboten för detta projekt fungerar i en stängd domän, alltså inom Siemens verksamhet, skulle den enbart kunna svara på frågor kring just den specifika domänen. På grund av vad supportteknikerna önskade för funktionalitet blev domänen ytterligare begränsad då den för detta projekt enbart är designad för att hantera en specifik fråga där användarna vill hitta information med ett visst taggnamn. Därmed måste chatboten be om specifik information från användaren för att underlätta användningen av chatboten och för chatboten att snabbt hitta rätt information. För eventuell framtida implementering bör dock ytterligare funktionalitet för chatboten diskuteras för att det ska vara meningsfullt att utveckla en chatbot.

Något ytterligare som denna prototyp inte hanterar är ifall det finns flera dokument som skulle kunna innehålla den relevanta informationen. Fördelaktigt borde chatboten ge flera dokument som förslag. Det enda som behöver designas om i detta fall är feedbackknapparna. Det går att ha kvar designen för dem som de är nu, men för att ge chatboten en mer specifik feedback kring vilket dokument informationen fanns i skulle designen för detta behöva ändras.

5.3 Framtida funktion och dagens teknik

Som tidigare redan nämnts är prototypen begränsad till att enbart hantera en typ av uppgift. Detta på grund av att det visades finnas problem med grundläggande funktioner, däribland att snabbt och smidigt kunna hitta rätt typ av information. Detta var den funktion och uppgift som ansågs kunna implementeras i en chatbot. Dock, för att få ut fulla potentialen av en chatbot bör den kunna hantera flera olika uppgifter, men som fortfarande är fokuserat på ett specifikt område. Därmed bör det undersökas mer kring vad för typ av uppgifter som denna chatbot skulle kunna hantera och hjälpa till med. Exempelvis skulle det kanske vara att hitta någon

annan typ av information, exempelvis att de även skulle kunna söka efter information på andra sätt än via taggnamn, site och dokumentnummer. En intressant infallsvinkel för framtida implementering skulle kunna vara att chatboten även ska kunna användas av noviser, alltså personer som inte har den kunskapsbas kring exempelvis taggnamn som supportteknikerna har idag. Därmed måste det undersökas hur sådan information ska efterfrågas till en chatbot och hur Siemens får användning av att noviser skulle kunna hitta denna information. Skulle fler kunna arbeta som supporttekniker utan att de behöver den kunskapsbas de har idag? Med fler uppgifter kommer chatbotens funktionalitet att få en större separation från att vara en vanlig sökmotor, vilket skulle skapa ytterligare argument för att en chatbot bör implementeras. Chatbotar är en typ av sökmotor, men den stora skillnaden ligger i hur användaren kommer åt informationen samt att informationssökningen sker i form av en konversation. Så därmed är det viktigt att tänka på hur informationen presenteras och hur konversationen sker.

Något som också bör undersökas mer är kring hur informationen ska återges till användaren av chatboten. I prototypen idag återges en länk till själva dokumentet som innehåller den information användaren söker. Dock borde eventuellt chatboten utvecklas på ett sådant sätt att den ger tillbaka mer specifik information, alltså att den skriver ut den information som eftersökes. Detta skulle underlätta för användarna att slippa leta efter specifik information i ett dokument. Dock behöver de ibland hela dokument då det är stora delar av det som är av relevans. Därför behöver detta undersökas mer, hur vill de få tillbaka informationen och i vilka fall vill de ha tillbaka informationen på detta sätt? Av sekretesskäl kan det även vara problematiskt att ge tillgång till hela dokument. Då det idag krävs olika typer av inloggningar och att de inte har behörighet till alla olika typer av dokument måste detta undersökas mer kring hur det ska gå till. Ska chatboten begära någon typ av inloggning för att veta vad för typ av behörighet just den personen har, eller ska inloggningen ske när dokumentet öppnas. Någon typ av funktion krävs för att begränsa hur användaren kommer åt dokumenten och den sökta informationen. Som Microsoft nämner i sin lista över designriktlinjer kan det vara bra att ta hjälp av ett säkerhetsteam för att vara säker på att data hanteras på korrekt sätt (Microsoft, 2018).

Utifrån de intervjuer som genomfördes samt den litteratur som samlats in anses en chatbot riktad mot kund eventuellt vara mer passande. Denna typ av chatbot skulle likna de chatbotar som finns idag där kunden kan ställa frågor till chatboten och få svar kring alla grundläggande frågor. Detta skulle leda till att supportteknikerna inte behöver lägga lika mycket tid på att hjälpa kunder med de lätta och repetitiva uppgifterna utan kan fokusera på de uppgifter som är mer

tidskrävande. Skulle därmed chatboten inte kunna besvara en fråga från användaren skulle den enkelt kunna skicka vidare frågan till en supporttekniker. Här är det då viktigt att tänka på den tredje designriktlinjen från Microsoft (2018) som handlar om att överlämnandet från en chatbot till en människa ska ske så smidigt som möjligt.

Vad för teknik som bör användas kring just denna chatbot är inte helt säkert, det beror helt och hållet på vad för typer av uppgifter den ska behandla. Är det enbart uppgifter som den prototyp som skapats i detta projekt hanterar finns det en möjlighet att använda sig av en regelstyrd chatbot, där det alltså är hårdkodat hur den ska bete sig. Dock är troligtvis maskininlärning en bättre teknik att använda sig av då detta är en teknik som fortsätter att utvecklas och kan lära sig hur den ska svara. Det uppfattas även vara bättre när det gäller att användarna ska få skriva in ett fritextsvar och få ett svar tillbaka som grundar sig i information som finns i dokumenten. Sedan vad gäller om NLP eller DL ska användas bör undersökas mer då denna begränsade introduktion som ges i denna uppsats inte kan leda till att ett välgrundat svar kan ges om exakt vilken som lämpar sig för just Siemens. Planeras det att chatboten ska implementeras senare kan DL eventuellt vara mer intressant då den har bidragit till att det gjorts stora framsteg inom AI, samt att det troligtvis kommer att forskas mer kring det framöver. Dock bör detta ses över när själva implementeringen ska ske för att den teknik som används ska passa för de uppgifter som chatboten ska utföra.

5.4 Möjlig framtida implementering

Den stora frågan som ställs utifrån denna studie som genomförts är ifall en chatbot är den rätta tekniken som söks för de typer av uppgifter som de vill att den ska hantera. För just den uppgiften som prototypen visar skulle lika gärna en sökfunktion kunna skapas som kan hantera och har tillgång till alla system som finns idag. En chatbot är dock väldigt lik en sökfunktion men att den blir något personligare och kan på ett enklare sätt presentera specifik information. Chatbotar är populär teknik att använda sig av idag, men det finns viss problematik med det som behöver övervinnas för att det ska vara lämpligt att implementera. Först och främst är det som tidigare nämnts viktigt att tänka på att chatboten ska lösa uppgifter på ett bättre sätt än om de som är tänkta att använda chatboten skulle utföra uppgiften utan chatboten. Den kommer inte att användas om den inte underlättar uppgiften. För att undersöka detta kan flera metoder kombineras som använts i denna undersökning. Bland annat skulle intervjuer eller enkäter kunna användas för att ta fram vad användarna förväntar sig av chatboten. Sedan skulle användartester tillsammans med intervjuer eller enkäter kunna genomföras för att användarna

ska få bedöma ifall chatboten lever upp till de förväntningar som de hade på den. Dessutom, för att det ska löna sig att utveckla och implementera en chatbot, bör den hantera flera uppgifter eller att det åtminstone finns planer på fler uppgifter som den ska hantera.

Det finns, som kan utläsas från tidigare diskussion, en hel del som måste utforskas vidare. Bland annat skulle användartester behöva utföras för att undersöka ifall prototypen är på rätt väg i vad supportteknikerna vill ha eller om den skulle behöva ändras på något sätt. Samtidigt behöver många andra delar utforskas ytterligare i form av bland annat ytterligare intervjuer med de tänkta användarna. Detta för att få en mer fokuserad och specificerad förståelse för just hur sökning av information går till och vad för information som är av relevans för dem. Sedan krävs det att det finns en stor kunskap i hur systemen är uppbyggda, hur de skulle kunna integreras och hur säkerheten kring dessa kan fungerar. Detta kommer vara essentiell information för att veta hur och vad för information som eftersökes kan presenteras. Genom dessa undersökningar kommer det att bli tydligare vad som bör implementeras och om chatbottekniken är det som bör användas. Det finns som tidigare nämnts stor potential kring chatbottekniken, så implementering av detta behöver inte vara omöjligt eller onödigt, men det krävs mer forskning och undersökning för att kunna understryka att det är det som bör införskaffas. Därmed kan det eventuellt vara fördelaktigt att inledningsvis implementera en sökmotor som tar hand om den uppgift och det problem som tas upp i resultatet från denna studie. Detta om de behöver implementera ett system som behöver hantera detta nu och inte vill invänta rätt teknik och underlag för att kunna implementera en chatbot.

6. Slutsats

Utifrån denna undersökning som genomförts har det uppkommit att det finns en del aspekter som bör undersökas ytterligare för att det ska vara möjligt och rimligt för Siemens att implementera en chatbot i deras system. Bland annat måste de se över hur de ska kunna integrera de olika systemen och hur säkerheten påverkar vad för typ av information som kan visas. De måste även undersöka eventuella fler uppgifter som chatboten kan utföra för att skapa en större mening med chatboten. Det blir en kostnadsfråga ifall det är värt att skapa en chatbot med bara en uppgift. Finns det en plan kring att utöka antalet uppgifter chatboten kan hantera finns det en större logik i att implementera en sådan i deras system även fast den i början enbart har ett användningsområde. Dock bör denna plan vara tydlig innan en implementering påbörjas. Så som det ser ut idag bör de eventuellt avvakta med att utveckla och implementera en chatbot tills att denna plan är uppsatt samt att de genomfört ytterligare undersökningar kring användningsområden och chatbottekniken.

Denna generella kunskap som kan förmedlas utifrån den undersökning som genomförts i detta projekt är att det krävs en grundlig undersökning i vad för användningsområde chatboten ska ha. Dessutom är det oerhört viktigt att den eller de uppgift(er) som chatboten ska genomföra ska ske på ett bättre, lättare och snabbare sätt än om chatboten inte använts. För att vara säker på att detta uppnås krävs det någon form av kravformulering för att veta vad för krav som man måste förhålla sig till. Det är även rekommenderat att genomföra en designprocess, likt i denna uppsats, med iterationer innan en implementation av en prototyp görs för att vara säker på att de funktioner som chatboten är tänkt att innehålla är designade på ett bra sätt. En prototyp eller även flera kan vara bra att skapa för att få interagera fysiskt med den design som tagits fram. Här blir det ännu tydligare vad som fungerar och inte, samt hur användarvänlig den faktiskt är. Det är just detta som är det viktigaste av allt. Är chatboten inte användarvänlig så kommer den inte användas. Finns det ens någon mening i att implementera en chatbot i verksamheten då? Det krävs en stor förståelse för både tekniken som finns i verksamheten och kring chatbotar, samt en stor kunskap och förståelse för användarna och vad de behöver. Detta för att kunna implementera en bra och användarvänlig chatbot som med större sannolikhet kommer att användas.

I och med att chatbottekniken är väldigt populär så kommer utvecklingen av den med stor sannolikhet att gå stadigt uppåt vilket leder till att det finns stor potential med att använda sig av chatbotar. Så det kan därmed vara ett väldigt intressant område att hålla ett öga på för

Siemens. Utför de grundliga undersökningar inom området, samt inom deras egen verksamhet, och gör upp en tydlig plan för vad för funktionalitet de behöver så finns det absolut möjligheter för Siemens att implementera en chatbot i deras system. Dock är det eventuellt inledningsvis en sökmotor som de behöver.

Referenser

Abu Shawar, B., & Atwell, E. (2003). Using dialogue corpora to train a chatbot. *Proceedings of the Corpus Linguistics 2003 Conference*, 681–690.

AISB. (n.d.). AISB - The Society for the Study of Artificial Intelligence and Simulation of Behaviour - Loebner Prize. Retrieved May 9, 2019, from http://www.aisb.org.uk/events/loebner-prize

Amershi, S., Weld, D., Vorvoreanu, M., Fourney, A., Nushi, B., Collisson, P., … Four-ney, A. (2019). Guidelines for Human-AI Interaction. *CHI 2019, May 4–9, 2019, Glasgow, Scotland Uk*, NA. https://doi.org/10.1145/3290605.3300233

Arvola, M. (2014). Interaktionsdesign och UX : om att skapa en god användarupplevelse (1st ed.). Lund: Studentlitteratur.

Arvola, M. (2019). *Interaktionsdesign och UX - om att skapa en god användarupplevelse* (2nd ed.). Linköping: Manuskript under förberedelse.

Batacharia, B., Levy, D., Catizone, R., Krotov, A., & Wilks, Y. (2013). CONVERSE: a Conversational Companion. *Machine Conversations*, 205–215. https://doi.org/10.1007/978-1-4757-5687-6_17

BBC NEWS. (2004). BBC NEWS | Technology | Alice chatbot wins for third time. Retrieved from http://news.bbc.co.uk/2/hi/technology/3672424.stm

Bermúdez, J. L. (2014). *Cognitive science : an introduction to the science of the mind* (2nd ed.). Cambridge: Cambridge University Press.

Biswas, M. (2018). Beginning AI Bot Frameworks: Getting Started with Bot Developement. Beginning AI Bot Frameworks. Kolkota, West Bengal: Apress.

Bocklisch, T., Faulkner, J., Pawlowski, N., & Nichol, A. (2017). Rasa: Open Source Language Understanding and Dialogue Management, 1–9. Retrieved from http://arxiv.org/abs/1712.05181

Carlshamre, P. (2001). A Usability Perspectiv On Requirements Engineering: From Methodology to Product Development. Linköping: Linköpings Universitet.

Chowdhury, G. G. (2005). Natural language processing. In *Finding the Concept, Not Just the Word* (pp. 67–78). https://doi.org/10.1016/b978-1-84334-318-9.50005-3

Colby, K. M. (1999). Human-Computer Conversation in A Cognitive Therapy Program. In *Machine Conversations* (pp. 9–19). Boston, MA: Springer US. https://doi.org/10.1007/978-1-4757-5687-6_3

De, B. (2017). API Management : An Architect's Guide to Developing and Managing APIs for Your Organization. Bangalore, Karnataka, Indien: Apress.

Dunn, T. (2019). Deep learning. *Salem Press Encyclopedia of Science*. Salem Press. Retrieved from https://login.e.bibl.liu.se/login?url=https://search.ebscohost.com/login.aspx?direct=true&AuthType=ip,uid&db=ers&AN=119214353&lang=sv&site=eds-live&scope=site

Fernandes de Mello, R., Antonelli Ponti, M., Fernandes de Mello, R., & Antonelli Ponti, M. (2018). A Brief Review on Machine Learning. In *Machine Learning* (pp. 1–74). https://doi.org/10.1007/978-3-319-94989-5_1

Goodwin, K. (2009). Designing for the digital age: How to create human-centered products and services. Indianapolis: Wiley Publishing, Inc.

Grudin, J., & Jacques, R. (2019). Chatbots, Humbots and the Quest for Artificial General Intelligence. *CHI Conference on Human Factors in Computing Systems Proceedings (CHI 2019)*, 11. https://doi.org/https://doi.org/10.1145/3290605.3300439

Holtzblatt, K., & Beyer, H. (2017). *Contextual Design*. https://doi.org/10.1016/b978-0-12-800894-2.00006-5

Io, H. N., & Lee, C. B. (2018). Chatbots and conversational agents: A bibliometric analysis. *IEEE International Conference on Industrial Engineering and Engineering Management*, 215–219. https://doi.org/10.1109/IEEM.2017.8289883

Iqbal, K., Berry, I., Spacil, L., Qian, C., & Standefer, R. (2019). Azure Bot Service Introduction - Bot Service | Microsoft Docs. Retrieved from https://docs.microsoft.com/en-us/azure/bot-service/bot-service-overview-introduction?view=azure-bot-service-4.0#building-a-bot

Luger, E., & Sellen, A. (2016). "Like having a really bad PA": the gulf between user expectation and experience of conversational agents. *CHI '16 Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, 5286–5297. https://doi.org/10.1145/2858036.2858288

Microsoft. (2018). Responsible bots: 10 guidelines for developers of conversational AI. Retrieved from https://azure.microsoft.com/en-

Olsson, L. (2019). AI-systems möjligheter i en avancerad support- och industrikontext. Linköpings Universitet.

Pandey, P. (2019). Building a Conversational Chatbot for Slack using Rasa and Python -Part 1. Retrieved from https://towardsdatascience.com/building-a-conversational-chatbot-for-slack-using-rasa-and-python-part-1-bca5cc75d32f

Phillips, C. (2018). Natural Language Processing (NLP) & Why Chatbots Need it. Retrieved from https://chatbotsmagazine.com/natural-language-processing-nlp-why-chatbots-need-it-a9d98f30ab13

Ratcliffe, L., & McNeill, M. (2012). *Agile experience design : a digital designer's guide to agile, lean, and continuous*. Berkley: New Riders. Retrieved from https://www.oreilly.com/library/view/agile-experience-design/9780132869249/

Russel, S., & Norvig, P. (2010). Artificial Intelligence: A Modern Approach (3rd edition). Pearson Education.

Shawar, B. A., & Atwell, E. (2007). Different measurements metrics to evaluate a chatbot system, (April), 89–96. https://doi.org/10.3115/1556328.1556341

Shawar, B. A., & Atwell, E. (2015). ALICE chatbot: Trials and outputs. *Computacion y Sistemas*, *19*(4), 625–632. https://doi.org/10.13053/CyS-19-4-2326

Siemens AB. (n.d.). Om Siemens - Om Siemens - Siemens. Retrieved May 7, 2019, from https://w3.siemens.se/home/se/sv/cc/om-siemens/Pages/om-siemens.aspx

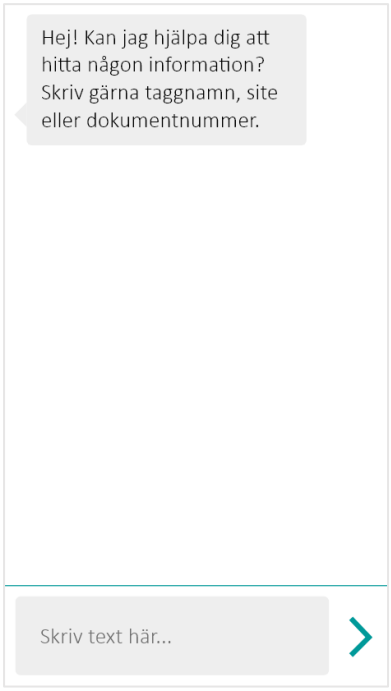
Velloso, M., Iqbal, kamran, & Standefer, R. (2017a). Design a bot's first user interaction - Bot Service | Microsoft Docs. Retrieved from https://docs.microsoft.com/en-us/azure/bot-service/bot-service-design-first-interaction?view=azure-bot-service-4.0

Velloso, M., Iqbal, K., & Standefer, R. (2017b). Principles of bot design - Bot Service | Microsoft Docs. Retrieved from https://docs.microsoft.com/en-us/azure/bot-service/bot-service-design-principles?view=azure-bot-service-4.0

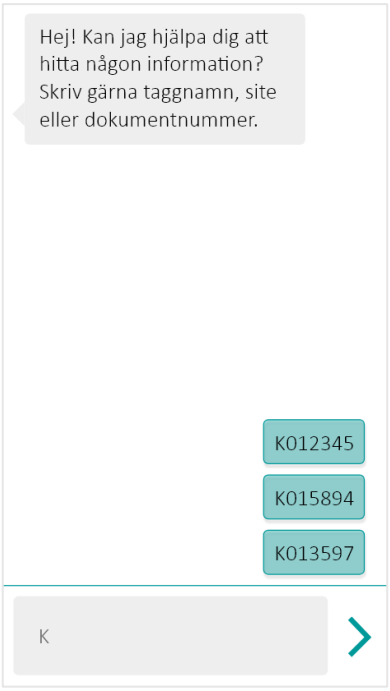
Weizenbaum, J. (1966). ELIZA---a computer program for the study of natural language communication between man and machine. *Communications of the ACM*, *9*(1), 36–45. https://doi.org/10.1145/365153.365168

Zamora, J. (2017). I'm Sorry, Dave, I'm Afraid I Can't Do That: Chatbot Perception and Expectations. *Proceedings of the 5th International Conference on Human Agent Interaction - HAI '17*, 253–260. https://doi.org/10.1145/3125739.3125766

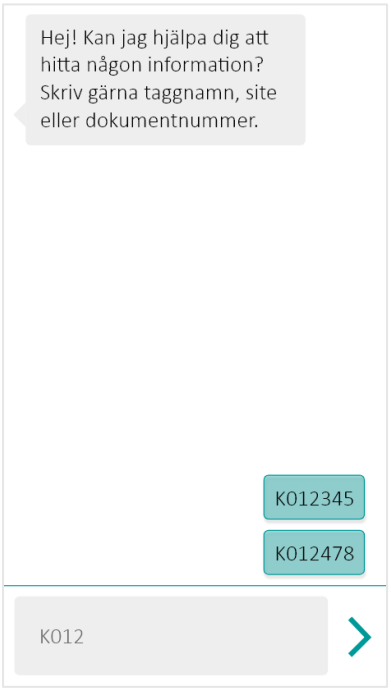
Appendix A – Fullständig prototyp, steg för steg



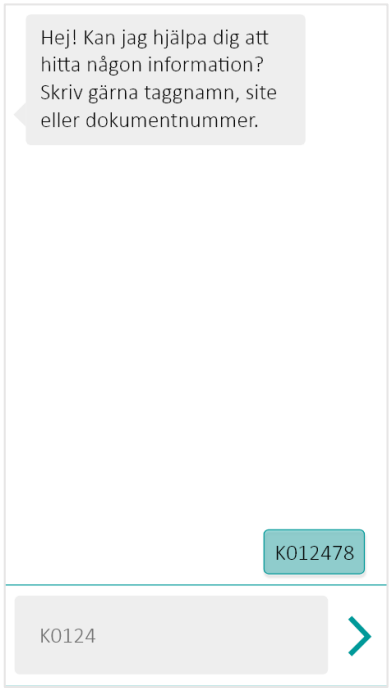
Steg 1. Chatboten ber om taggnamn, site eller dokumentnummer.



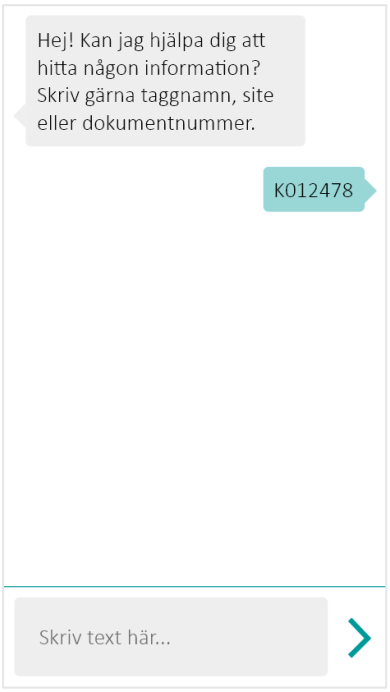
Steg 2. Användaren börjar skriva ett taggnamn.



Steg 3. Ju mer användaren skriver desto färre blir förslagen.



Steg 4. Tills det enbart finns ett förslag kvar, om användaren inte tryck på ett förslag innan.

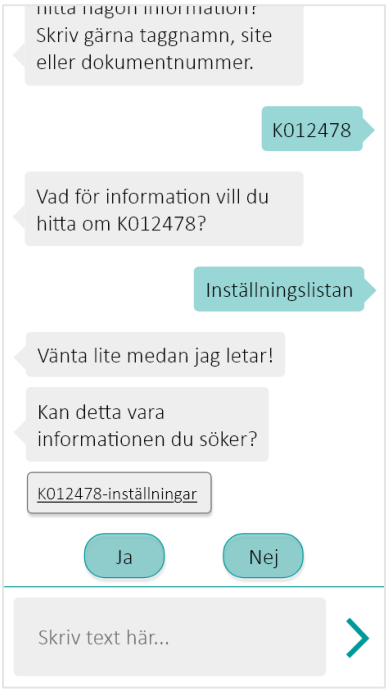


Steg 5. Det valda förslaget skickas som svar till chatboten.

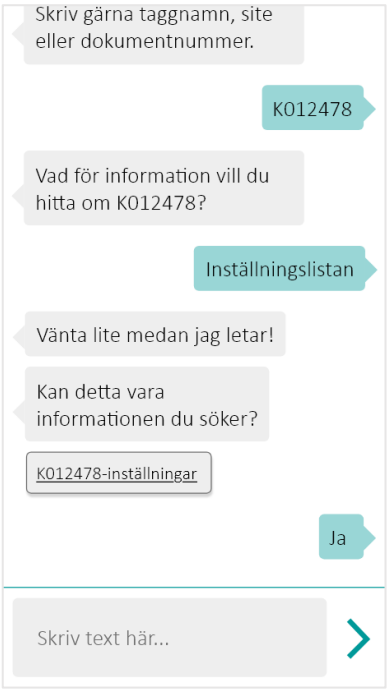
Steg 6. Chatboten frågar vad för mer specifik information användaren eftersöker.

Steg 7. Användaren svarar i fritext.

Steg 8. Chatboten ber användaren att vänta medan chatboten letar.

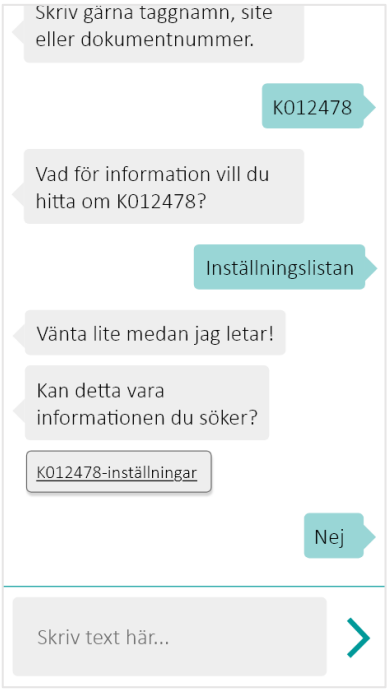


Steg 9. Chatboten förser med förslag på dokument med den sökta informationen och ber användaren svara på om det var rätt eller fel.

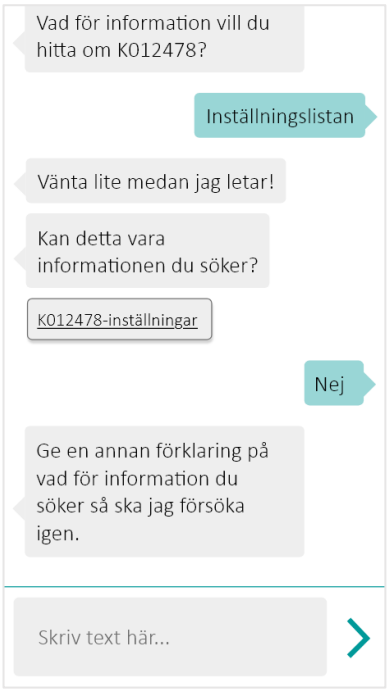


Steg 10a. Om användaren trycker på ja skickas detta som svar till chatboten.

Steg 11a. Chatboten tackar för svaret.



Steg 10b. Om användaren trycker nej skickas detta som svar till chatboten.



Steg 11b. Chatboten ber användaren om ny information.