Undersøgelse af designstrategi for forskellige programmeringssprog

Af Jonas S. Walsted - Gruppe 1083 - Studie nr. 20160821

MSc. i Produkt- og designpsykologi
4. semester
Aalborg Universitet

AALBORG UNIVERSITET
STUDENTERRAPPORT

Andet Studieår v/ Produkt- og designpsykologi
Produkt- og designpsykologi
Fredrik Bajers Vej 7A
9220 Aalborg
https://www.es.aau.dk/

Titel:

Undersøgelse af designstrategi for forskellige programmeringssprog.

Tema:

Kandidatspeciale

Projektperiode:

Februar - Juni 2022

Studerende:

Jonas S. Walsted

Studie nr.:

20160821

Mail:

jwalst16@student.aau.dk

Semester:

4. Semester

Vejleder:

Rodrigo Ordoñez

Sidetal: 96
Appendiks: 60
Afsluttet 01-06-2022

Abstract:

There has been a marked increase in the number of third-party libraries and their complexity. Furthermore, there are some common challenges in the documentation of the code. Therefore, this report aims to investigate how to develop documentation for third-party help libraries so that the work patterns of experts and beginners are taken into account. Furthermore, the project wants to investigate how to include several programming languages in the same documentation. Next, the project was limited to examining the design and structure of a front-page, how-to guide, and setup guide. Furthermore, the starting point was the WappstoIoT library, which Seluxit developed. The project first gathered knowledge about the target group through existing literature and then via interviews and online questionnaires. Based on this knowledge, five iterations were performed, which among other things, combined heuristic evaluation and usability testing. Based on the results from the last iteration, it was demonstrated that it successfully designed the focus pages that fulfilled the project's purpose with the WappstoIoT library.

Forord

Denne rapport er udarbejdet af Jonas Søvind Walsted, som er 4. semesters studerende på kandidateuddannelsen Produkt og Designpsykologi ved Aalborg Universitet.

Projektets formål er, at undersøge hvordan dokumentation til trejdeparts biblioteker kan udarbejdes på en sådan måde, at udviklere på alle niveauer, kan benytte det. Emnet er valgt ud fra et projektforslag, der blev opstillet af Seluxit A/S[1].

Seluxit er en Internet of Things virksomhed, som har udviklet flere trejdeparts biblioteker til deres IoT plarform Wappsto[2]. Undervejs har der været faglig sparring med ansatte ved Seluxit A/S i forbindelse med designudviklingen, hvor personale også er blevet inddraget til f.eks. workshop, evalueringer og lignede. Uanset samarbejdet med Seluxit har formålet med projektet ikke været at udvikle dokumentation passende til Seluxit A/S specifikt, men derimod at udvikle et generisk design, der kan anvendes af en vilkårlig producent med et lignede størrelse trejdeparts bibliotek.

Jeg vil gerne takke særligt Daniel Lux for at stille sig selv og ansatte ved Seluxit A/S til rådighed under udarbejdelsen af projektet, samt for at have stillet lokaler til rådighed under projektet. Ligeledes vil jeg gerne takke alle ansatte ved Seluxit for at stille sig til rådighed ved tvivlsspørgsmål, workshops, interviews og lignede.

Læsevejledning

Projektet indeholder en hovedrapport og dertil et appendiks, som kan findes efter litteraturlisten. Derudover er et bilag vedhæftet som en ZIP-fil. En oversigt af bilagene kan findes i appendiks H.

Til henvisning af kilder i rapporten benyttes der Harvard metoden, derfor refereres kilderne først med efternavn og dernæst med år.

Litteraturlisten, som referencerne henviser til kilderne med forfatter, titel, udgave, udgivelsesår og forlag. Internetsider angives yderligere med link.
Der vil ligeledes være figurer og tabeller i rapporten, som er nummereret i henhold til, hvilket kapitel de er placeret i, samt i nummeret rækkefølge.

[1] `https://www.seluxit.com/`
[2] `https://wappsto.com/`

Indholdsfortegnelse

Kapitel 1 Introduktion **1**
1.1 Dokumentation . . . 1
1.2 Seluxit A/S . . . 4
1.3 Projekts forløb . . . 5

Kapitel 2 Udvælgelse af biblioteket **6**
2.1 Proces for indsamling af data omhandlende målgruppen . . . 6
2.2 Forskelle i målgruppen . . . 6
2.3 Design for eksperter og nybegyndere . . . 9
2.4 Faktorer der influerer i valget af softwarebiblioteker . . . 11
2.5 Problemformulering . . . 12
2.6 Afgrænsning . . . 12

Kapitel 3 Brugerundersøgelser **13**
3.1 Semistruktureret interviews . . . 13
3.1.1 Introduktion . . . 13
3.1.2 Resultater . . . 15
3.1.3 Diskussion . . . 23
3.1.4 Opsummering af væsentlige resultater . . . 24
3.2 Spørgeskema . . . 25
3.2.1 Introduktion . . . 25
3.2.2 Resultater . . . 26
3.2.3 Diskussion . . . 36
3.2.4 Opsummering af væsentlige resultater . . . 37
3.3 Konklusion på brugerundersøgelsen . . . 38

Kapitel 4 Udformning af prototype **40**
4.1 Design studio workshop . . . 40
4.1.1 Resultater . . . 43
4.1.2 Diskussion . . . 45
4.1.3 Delkonklusion . . . 47
4.2 1. version . . . 47
4.2.1 Forside . . . 48
4.2.2 Get Started . . . 50
4.2.3 Working Example . . . 52
4.3 Heuristik Evaluering . . . 55
4.3.1 Resultater . . . 56
4.3.2 Diskussion . . . 57
4.3.3 Delkonklusion . . . 58
4.4 2. Version . . . 58
4.4.1 Forside . . . 59

4.4.2 Get Started og Quick Start . . . 61
4.4.3 Working Example . . . 62
4.5 Feedback fra udviklere . . . 64
4.5.1 Resultater . . . 65
4.5.2 Diskussion . . . 67
4.5.3 Delkonklusion . . . 68
4.6 3. Version . . . 68
4.6.1 Forside . . . 69
4.6.2 Get Started . . . 70
4.6.3 How to Guide . . . 70

Kapitel 5 Dataindsamling fra målgruppen 72
5.1 Forsøgsdesign . . . 72
5.1.1 Forsøgspersoner . . . 73
5.1.2 Fremgangsmåde . . . 73
5.2 Pilotforsøg . . . 74
5.2.1 Resultater . . . 74
5.2.2 Diskussion . . . 75
5.2.3 Delkonklusion . . . 75
5.3 3.5 version . . . 76
5.3.1 Ændringer . . . 76
5.3.2 Resultater . . . 78
5.3.3 Diskussion . . . 80
5.3.4 Delkonklusion . . . 81
5.4 4. version . . . 82
5.4.1 Ændringer . . . 82
5.4.2 Resultater . . . 83
5.4.3 Diskussion . . . 85
5.4.4 Delkonklusion . . . 86
5.5 5. Version . . . 86
5.5.1 Ændringer . . . 86
5.5.2 Resultater . . . 87
5.5.3 Diskussion . . . 90
5.5.4 Delkonklusion . . . 91

Kapitel 6 Diskussion 92

Kapitel 7 Konklusion 95

Litteratur 97

Appendiks A Interviewguide 100
A.1 Interviewguide . . . 100

Appendiks B Udvikling af spørgeskemaet 104
B.1 Resultater - Samlet . . . 105
B.2 Svarfordeling for Deltidsudviklere og Karriereudvikler . . . 121

Appendiks C Heuristisk Evaluering **127**
C.1 Forsøgsopstilling 127
C.2 Referat 127

Appendiks D Feedback fra udviklere **135**
D.1 Forsøgsopstilling 135
D.2 Referat fra feedback mødet 135

Appendiks E Kodeeksempler **141**
E.1 Python 142
E.2 JavaScript 145
E.3 Arduino 146

Appendiks F Usability test **148**
F.1 Information om forsøgspersoner 148
F.2 Introduktion 149

Appendiks G 5 version - Gennemgang af strukturen **150**
G.1 Python 154
G.2 JavaScript 157

Appendiks H Bilagoversigt **160**

1 Introduktion

I den virkelige verden er biblioteker steder, hvor du kan få adgang til information via f.eks. bøger og artikler. Men hvad angår programmering, er et bibliotek en samling af forudskrevet kode, som kan indeholde objekter, funktioner og metoder, som kan genbruges og deles imellem udviklere. Formålet med bibliotekerne er generelt, at udføre specifikke, veldefinerede operationer og derfor udvikles de normalt, når et problem opstår hyppigt med et bestemt domæne [Nikola Kolev, 2017].

Eksempler på populære biblioteker er JavaScript biblioteket Chart.js[1], som bruges af designere og udviklere til at tilføje grafer til deres hjemmesider. Et andet eksempel er Numpy[2], som er et Python biblioteket, som blandt andet bruges indenfor *machine learning*.

Brugen af softwarebiblioteker er steget markant de seneste år. Ifølge Kula German [2018] havde Maven[3] 260.000 biblioteker i 2010, men i 2016 var det antal oppe på 1.669.639 unikke Maven biblioteker. Dette gør Maven til et af de største platforme af *repositories* for *open source* software biblioteker i verden. En anden platform, som også har oplevet en væsentlig stigning af *repositories* og brugere er Github[4]. Ifølge Jason Warner [2018], er GitHub gået fra 33.000 *repositories* i 2008 til over 100 millioner i 2018, hvor der er ca. 31 millioner udviklere tilknyttet platformen.

1.1 Dokumentation

For at brugere kan benytte sig af biblioteker skal disse dokumenteres. Via dokumentationen kan brugerne finde viden om f.eks. hvordan man installere og inddrager biblioteket i sit eget projekt.

Toby Osbourn [2019] beskriver i sin artikel, at der er mange fordele ved at ens kode er veldokumenteret. En af de større fordele er, at man giver brugere muligheden for at hjælpe sig selv ved tvivlsspørgsmål og derfor ikke har behov for support. At holde dokumentationen til et trejdspart biblioteker åben, vil ydermere hjælpe søgemaskiner såsom Google med at indeksere siderne. Derfor vil en veldokumenteret kode også hjælpe til kendskabet til ens virksomhed fra udviklerne perspektiv.

Parnas [2003][s. 125] forklarer, at en generel problemstilling med brugerdokumentation er, at udarbejdelsen af dokumentationen ofte bliver overladt til tekniske skribenter, som ikke nødvendigvis kender til alle detaljerne i systemet. Derfor kan brugerdokumentationen ofte

[1] `https://www.chartjs.org/`
[2] `https://numpy.org/`
[3] `https://maven.apache.org/`
[4] `https://github.com/`

være misvisende, inkonsistente og ikke færdiggjorte. Disse ting kan medføre, at brugerne finder dokumentationen dårlig organiseret og forberedt, hvilket gør de ikke vil bruge biblioteket.

Nogle udviklere har forslået, at informationer om f.eks. funktioners virken, skal placeres i selve koden. Her argumenterer Parnas [2003][s. 128] dog for, at selvom det vil hjælpe udviklerene selv, vil det ikke opfylde andre brugers behov og det bør ikke være nødvendigt at være i koden, for at finde ud af hvordan man bruger den.

Ifølge [Geiger et al., 2018][s. 775] er et af de større problemer indenfor dokumentation af software, at der er uklarhed og manglende konsistens omkring definitioner og typer af dokumentation. Geiger et al. [2018] fandt via sine undersøgelser frem til seks forskellige former for dokumentation, som alle seks har forskellige formål, målgruppe og kræver forskellige forståelsesniveau for at benytte.

En af disse typer af dokumentation er brugerdokumentation. Målgruppen for denne type dokumentation er brugeren af bibliotekerne. Brugerdokumentationen har til formål, at guide brugeren i hvordan man installere biblioteket og hvordan bibliotekets funktioner fungerer på et overordnet niveau. Brugerdokumentation indeholder ligeledes setupguides, tutorials og *getting started*-guides. En anden type af dokumentation er *Galleries and examples*, som igen er henvendt til brugeren, men indeholder *scripts*, der beskriver, hvordan man udfører handlinger ved hjælp af softwarebiblioteket. Application Programming Interface dokumentation, henvender sig også til brugeren af biblioteket, men bringer en mere detaljeret beskrivelse af hvordan funktioner, argumenter og parametre kan bruges i biblioteket. Denne type kan også indeholde kode snippets og vil typisk være tekst inkluderet i kodekommentarer i begyndelsen af funktioner eller metoder. Typen utraditionel dokumentation dækker over fejlbeskeder, socialemedier platforme, Q&A hjemmesider og ad-hoc instrukser, der omhandler brugen af biblioteket. Denne type dokumentation henvender sig til brugeren af biblioteket. Udviklerdokumentation er en type af dokumentation, der henvender sig til de folk der bidrager til biblioteket. Denne type indeholder retningslinjer for bidrag og teknisk information om projektet, såsom automatiske byggesystemer til test af kode. Den sidste type af dokumentation som Geiger et al. [2018] fandt frem til var kommentarer i koden. Denne type henvender sig til udviklere, der bidrager til biblioteket og indeholder forklaringer af, hvordan bibliotekets kode fungerer.

Parnas [2003][s. 129] har en anden tilgang til opdelingen af dokumentationen, da Parnas [2003] opdeler efter arbejdsmønstret med dokumentationen. I artiklen bliver det beskrevet, at dokumentation findes i to former, enten som en guide eller som et opslagsværk. Her vil guiden typisk have til formål, at man som bruger læser alt fra start til slut og udfører handlingerne slavisk. Ydermere vil guiden primært henvende sig til den uerfarne udvikler, som har lidt til ingen erfaring med domænet. Opslagsværket derimod, er mere henvendt til udviklere, der allerede ved en del omkring biblioteket og har behov for at udlede specifik information. Årsagen til, at der findes disse to former for dokumentation, er ifølge Parnas [2003][s. 129], at i software feltet har uerfarne udviklere behov for en grundlæggende forståelse for domænet, hvor mere erfarne udviklere har behov for reference dokumenter.

Kvalitet af dokumentation

Ifølge Bhatti et al. [2021][s. 148] har medarbejdere hos Google defineret, at dokumentationen af software er god, hvis den opfylder sit formål. Formålet med dokumentationen bør stemme overens med formålet med ens kode, som blandt andet er at påvirke specifik brugeradfærd. Google opdelte herefter dokumentationskvaliteten i to grundlæggende kategorier; Funktionel kvalitet og strukturel kvalitet.

Med funktionel kvalitet menes der, om dokumentationen opnår sit mål og undersøger derved på et grundlæggende niveau, om dokumentet virker eller ej. Funktionel kvalitet bliver af Bhatti et al. [2021][s. 149 - 154] delt op i fem underkategorier. Den første underkategori er, hvor tilgængeligt dokumentationen er for slutbrugeren, hvori der inkluderes sprog, læseniveau og adgang til dokumentationen. Den anden underkategori er, hvor målrettet ens dokumentation er. Her forklarer Bhatti et al. [2021][s. 152], at man bør formidle formål med biblioteket allerede i titlen og i første afsnit af dokumentationen. Den resterende del af indholdet bør herefter opfylde dette formål. Den tredje underkategori er, hvor let slutbrugeren kan finde dokumentationen og navigerer rundt i den. Den fjerde underkategori er, hvor nøjagtig ens dokumentation er. Her vil dokumentation med høj nøjagtighed blandt andet have korrekt og opdateret tekniske forklaringer af koden. Den femte og sidste kategori er, hvor komplet ens dokumentation er. For at ens dokumentation kan være komplet skal den indeholde alle de oplysninger, der er nødvendige for, at slutbrugeren kan få succes med sit mål. Ud fra disse fem underkategorier ses det at funktionel kvalitet kan være svært at måle holistisk, men bør mere måles metrisk, da det hænger sammen med dokumentets formål [Bhatti et al., 2021][s. 148]

Den strukturelle kvalitet beskriver, hvor godt dokumentet er skrevet og hvor let det er for slutbrugeren at læse. I denne kategori inkluderer sætnings-, afsnits- og overskriftsstruktur, sprogets kvalitet og grammatikkens nøjagtighed. Denne kategori inddeles i tre underkategorier af Bhatti et al. [2021][s. 155 - 156]. Den første underkategori er, hvor nemt ens dokumentation er at forstå. Med dette menes der, hvor let læseren kan følge oplysningerne fra dokumentationen og hvor sikre brugeren er på, at de vil lykkes. Den anden underkategori er kortfattethed, med dette menes der at dokumentationen kun skal indeholde information, der er relevant for slutbrugerens mål. Den tredje underkategori er, hvor konsistens dokumentationen er. Dokumentationen bør være konsistens i sin struktur, sit sprogbrug, samt de begreber der bliver brugt igennem dokumentationen.

Bhatti et al. [2021][s. 158] forklarer, at ideelt set bør dokumentationen være både af høj strukturel og funktionel kvalitet. Men ifølge Bhatti et al. [2021] er funktionel kvalitet vigtigere, da velstruktureret dokumentation, der ikke når sit mål, er er værre for brugeren end dokumentation med strukturelle problemer, der stadig opfylder sit mål.

Kendte problemstillinger

Satish and Anand [2016] har undersøgt, hvilke problemstillinger der er almindelige for dokumentation af software. Dette er gjort ved gennemgå 30 forskningsartikler, der hver især har undersøgt dette emne med forskellige vinkler. Satish and Anand [2016]'s artikel tager udgangspunkt i software dokumentation af kode der bruges internt i virksomheder og kode der skal overleveres til eksterne parter. Derfor er denne artikel ikke rettet udelukkende

mod trejdeparts biblioteker, som dette projekt omhandler. Dog vurderes det stadig at have relevans, at undersøge hvilke faktorer der indspiller sig ved udviklingen af dokumentation på et generelt plan.

Satish and Anand [2016] konkludere ud fra dennes artikel, at der findes nogle gængse udfordringer, når det kommer til dokumentationen af kode. Satish and Anand [2016][s. 5] konkluderer blandt andet, at dokumentationen til software ofte er forældet. En anden almindelig problemstilling er, at der ikke findes nogle standarder for dokumentation og at udviklere ofte skal bruge meget tid på at opdatere sin dokumentation. Satish and Anand [2016][s. 5] forklarer ydermere, at udviklere ofte udviser en manglede entusiasme for arbejdet med dokumentation.

Baseret på analysen, har Satish and Anand [2016][s. 6] udarbejdet en række anbefalinger om softwaredokumentation. Her mener Satish and Anand [2016] blandt andet, at dokumentationens indhold og størrelse skal være afhængig af projektets størrelse, samt slutbrugerens erfarings- og kompetenceniveau.

En problemstilling som Bhatti et al. [2021][s. 3] påpeger ift. dokumentation til slutbrugere er, at udviklere generelt har en manglende viden omkring, hvilke informationer der er relevante for slutbrugeren. En årsag til dette kan være *the curse of knowledge*, som fremkommer, når en udvikler har siddet i lang tid med et system eller bibliotek og derfor forventer at visse emner er implicitte. Bhatti et al. [2021][s. 3] forklarer, at for at bryde *curse of knowledge*, kræver det at udviklerene forstår brugerens mål og evner indenfor det påkrævet emne.

Parnas [2003][s. 125] argumentere for, at udviklere generelt har en manglende forståelse for, hvordan man kan udarbejde dokumentation. Udover den manglende viden kan en anden årsag ifølge Parnas [2003][s. 133] være, at det kan være svært at måle på effekten af ikke at producere veldokumenteret dokumentation for sit bibliotek og derfor bliver dette nedprioriteret af mange. Grunden til denne nedprioritering kan være på grund af, at konsekvenserne ved dårlig dokumentation er inddirekte som f.eks. forsinkelser og tid udviklere bruger på at finde fejl i koden. Thoriq Firdaus [2021] underbygger denne påstand ved at forklare, at en anden grund til at der generelt bliver lavet mangelfuld dokumentation, er fordi udviklere har en tendens til at antage at brugeren har samme programmeringsfaglige niveau som udvikleren selv.

1.2 Seluxit A/S

En virksomhed som har udviklet trejdepartsbiblioteker er Seluxit. Seluxit er en *Internet of Things* (herefter IoT) virksomhed, som siden sin grundlæggelse i 2006, primært har beskæftigede sig med IoT løsninger indenfor Smart Meter, Smart Homes og Smart Machines [Seluxit A/S, 2021].
Seluxit har til dette formål udviklet deres egen IoT platform kaldet Wappsto[5]. Hensigten med Wappsto er blandt andet, at man som bruger kan sende og modtage data, samt automatisere handlinger for sine enheder. Et eksempel på dette kan f.eks. være at hvis man som bruger har udført et hjemmeprojekt med en Arduino, som kan åbne og lukke

[5] `https://wappsto.com/`

for varmen. Så kan man ved at koble Arduino'en til Wappsto fjernstyre temperaturen, samt automatisere varmereguleringen baseret på f.eks. vejrmeldinger eller varme priser. Tredjeparts biblioteket, der gør dette muligt hedder WappstoIoT[6] og er udviklet til programmeringssprogene Python, JavaScript og C/C++.

Under et initierende møde udtalte medarbejdere fra Seluxit, at de mener at deres dokumentation af WappstoIoT er mangelfuld og dette gør at eksterne bruger ikke benytter deres biblioteker. I udarbejdelsen af dokumentationen til Seluxits trejdeparts biblioteker, har det været udviklerne selv, der har udarbejdet dokumentationen til WappstoIoT. Bibliotekerne er blevet revideret af kollegaer, men selve målgruppen for bibliotekerne er ikke blevet inddraget.

1.3 Projekts forløb

Med det gennemgået teori in mente, vil dette projekts formål være at undersøge, hvordan man kan inkludere både de erfarne og uerfarnes arbejdsmønstre i samme dokumentation. Ydermere ønsker projektet at undersøge, hvorledes man kan inkludere flere programmeringssprog i samme dokumentation.

Projektet vil have fokus på brugerdokumentation til hjælper-biblioteker. Det vælges det at tage udgangspunkt i Seluxit's problemstilling. Selvom der tages udgangspunkt i Seluxits problemstilling, ønskes det at udarbejde en generisk løsning, som kan overføres til lignede problemstillinger.

[6] `https://github.com/wappsto`

Udvælgelse af biblioteket 2

Dette kapitel har til formål at indsamle viden fra eksisterende litteratur omkring emnerne der findes relevante i udviklingen af dokumentation for trejdeparts biblioteker.

2.1 Proces for indsamling af data omhandlende målgruppen

For at kunne udarbejde effektiv dokumentation, kræver det ifølge Bhatti et al. [2021][s. 21], at man har opnået forståelse for sine brugeres mål og behov. Dette kan opnås via brugerundersøgelser som f.eks. interviews og spørgeskemaer.

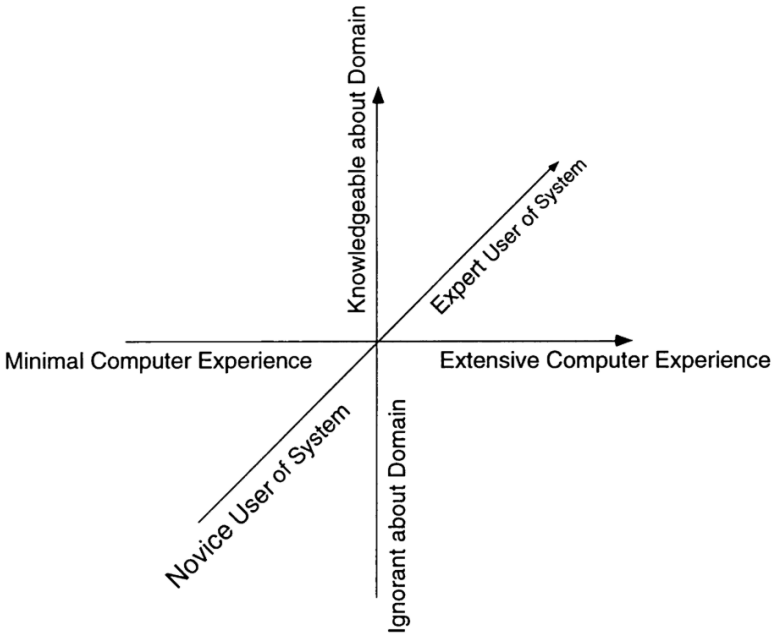
Dog bør man først forsøge at konkretisere hvem ens brugere er og hvad deres mål med dokumentationen er [Bhatti et al., 2021][s. 4 - 5]. I dette projekt vil målgruppen som udgangspunkt være udviklere på begyndeniveau til udviklere der arbejder med programmering til dagligt. Deres mål med dokumentationen vil afhænge af biblioteket, dog tages der i dette projekt udgangspunkt i WappstoIoT biblioteket. Derfor kan målet være at koble sine enheder op til internettet for at automatisere handlinger (se afsnit 1.2).

En metode, der kan hjælpe til at validere ens antagelser om målgruppen er interviews. Interviews er en metode, som har den fordel, at man kan stille åbne og uddybende spørgsmål til sin målgruppe. Når man skal udføre interviews skal man overveje for om de personer, der bliver interviewet er repræsentativ for målgruppen [Bhatti et al., 2021][s. 13]. Det anbefalet antal af personer man bør interviewe varierer, men ifølge Bhatti et al. [2021][s. 12] er tre til fem personer nok til at kunne skabe grundlag for fremtidige beslutninger.

Hvis man har behov for kvantitativ data for at forstå brugerens behov, kan spørgeskemaer ifølge Bhatti et al. [2021][s. 13] være fordelagtige. Spørgeskemaer kan ydermere være fordelagtige at bruge, hvis man ønsker at indsamle empiri fra en større del af populationen end interviews som udgangspunkt giver mulighed for.

2.2 Forskelle i målgruppen

Dette afsnit har til formål at belyse de faktorer, der udgør forskelle på hvordan brugere benytter et system.



Figur 2.1. De tre hoveddimensioner, som brugernes erfaring adskiller sig: Generelt viden om computere, ekspertise i at benytte et specifikt system og forståelse af opgavedomænet. Taget fra Nielsen [1993][s. 44]

Ifølge Nielsen [1993][s. 43 - 47] findes der tre dimensioner der adskiller brugerne. Disse tre kan ses på figur 2.1.

En dimension, der adskiller brugernes niveau er brugerens domæne kendskab. Til dette forklarer Nielsen [1993][s. 46], at en af forskellene er, at hvis man designer en brugergrænseflade til domæneeksperter, kan man benytte specialiseret terminologi og vise flere informationer ad gangen end tilfældet er med brugere med ringe domænekendskab. Brugere der har ringe domænekendskab har behov for at systemet forklarer, hvad systemet kan og gør, samt hvad de forskellige funktioner betyder. Den anvendte terminologi må ydermere ikke være forkortet, som tilfældet er for brugere med omfattende domæneviden.

En anden dimension der adskiller brugerne er deres generelle ekspertise indenfor computere. Her nævner Nielsen [1993] blandt andet, at brugere med erfaring fra andre lignede systemer, normalt vil være bedre til at løse problemstillinger end brugere med ringe computerforståelse. Dette er grundet, at den erfarne bruger vil benytte deres erfaring fra lignede systemer til at løse problemstillingen.

Den tredje dimension, er brugerens ekspertise i at benytte et specifikt system. Denne dimension er ifølge Nielsen [1993][s. 43], den dimension der normalt refereres til, når der diskuteres brugernes erfaring. Her nævner Nielsen [1993], at bruger kan være nybegyndere eller eksperter af et system eller på et niveau imellem disse to. Når en nybegynder udforsker nye systemer benytter de sig, ifølge Shneiderman [2004][s. 533], af erfaringer fra den rigtige verden og erfaringer fra andre brugergrænseflader, samt af gætværk. Grundet dette vil det være sjældent set, at en nybegynder vil forsøge at læse dokumentation fra ende til anden, inden de vil forsøge at gøre noget aktivt. Her referer Shneiderman [2004] til "Active user paradox", der siger at brugerens iver for at udarbejde noget meningsfyldt, stopper dem ofte i at bruge tid på at lære systemet at kende. Disse observationer har blandt andet

udmundet sig i *minimal manuals*, der opfordre til aktiv involvering med hands-on erfaring så hurtigt som muligt i dokumentationen. Ydermere opfordre denne type manualler til at dokumentationen kun beskriver nødvendige informationer for at en bruger kan udføre sine opgaver. Her beskriver Shneiderman [2004][s. 532 - 533], at komplette eksempler af opgaver er behjælpelige for brugeren til, at give et indblik i de funktionaliteter et system indeholder. Grunden til dette er, at mange brugere vil arbejde sig igennem disse eksempler for at verificere sin forståelse af systemet, samt opnå mere viden om det.

Ifølge Proctor and Zandt [2008][s. 318] udfører folk opgaver på forskellige måder, afhængigt af deres færdighedsniveau. Efterhånden som ekspertise erhverves, skifter folk fra én måde at udføre en opgave på til en anden. F.eks. opdeler FITTS and POSNER [1967] processen for at lære et domæne at kende i tre faser. Første fase, hvor personen er mindst dygtig kaldes den kognitive fase. Dernæst kommer associative fase og til sidst den autonome fase.

Ydeevnen i den indledende kognitive fase er bestemt af, hvor godt instruktioner og demonstrationer er givet. Her forsøger uerfarne brugere stadig at forstå systemet og derfor skal den uerfarne håndtere signaler og begivenheder i brugergrænsefladen, som ikke vil kræve opmærksomhed i senere faser. I den associative fase begynder de opgavekomponenter, der er lært i den kognitive fase, at blive relateret til hinanden. Dette opnås ved at kombinere disse komponenter i enkelte procedurer. Den endelige autonome fase er karakteriseret ved automatiseringen af disse procedurer, hvilket gør dem mindre underlagt kognitiv kontrol.

En anden ændring, der afhænger af personen færdighedsniveau er adfærden ved operation af systemet. Rasmussen [1986][s. 100] opdeler, hvorledes personen reagerer i specifikke situationer i tre niveauer. Disse niveauer kaldes færdigheds-, regel- og vidensbaseret adfærd. Proctor and Zandt [2008][s. 319 - 320] forklarer at færdighedsbaseret adfærd involverer simple relationer mellem stimuli og reaktioner. Opgaveydelsen bestemmes på dette niveau af automatiske handlinger, der ikke kræver bevidst kontrol fra personens side. Dette niveau vil derfor indeholde rutineaktiviteter under velkendte omstændigheder.
Den regelbaseret adfærd styres af regler eller procedurer lært fra tidligere erfaringer eller gennem instruktioner. Dette niveau af adfærd opstår, når personen oplever en afvigelse fra de planlagte eller forventede handlinger. Regelbaseret adfærd er målorienteret og typisk under bevidst kontrol. Videnbaseret adfærd bruges derimod i situationer, hvor ingen kendte regler er gældende. Vidensbaseret adfærd er afhængig af en konceptuel model for domænet. På dette niveau skal personen formulere et konkret mål og derefter udvikle en brugbar plan for det. Den vidensbaseret adfærd involverer problemløsning, ræsonnement og beslutningstagning. Ifølge Proctor and Zandt [2008][s. 319] vil selv en erfaren bruger vende tilbage til tidligere kontrolniveauer under visse omstændigheder. Dette bliver understøttet af Nielsen [1993][s. 45], der argumenterer for, at selvom en brugere benytter et system ofte, vil de fleste bruger aldrig opnå ekspertise i alle dele af et system. Grunden til dette er, at næsten alle systemer, der har en vis kompleksitet, har så mange funktioner og anvendelsesmuligheder, at brugeren normalt kun gør omfattende brug af en lille del. På grund af dette, kan en bruger der ellers anses som ekspert i dele af systemet være på begynderniveau i andre dele af systemet.

Selvom der forskelle i, hvordan erfarne og uerfarne brugere tilgår en opgave argumentere Proctor and Zandt [2008][s. 325], at eksperter ikke er i stand til at gøre ting, som

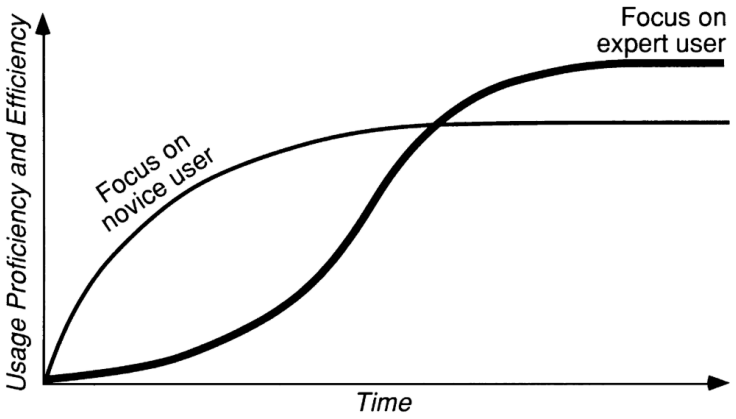
nybegyndere ikke kan. Dog besidder eksperter en let tilgængelig samling af fakta og procedurer, der anvendes, når der opstår problemstillinger i deres pågældende domæne. Dette betyder, at eksperternes særlige evner hovedsageligt stammer fra den betydelige mængde viden, de har om et bestemt domæne, og ikke fra en mere effektiv generel evne til at tænke. Proctor and Zandt [2008][s. 326] argumenterer for, at fordi eksperter har bedre mentale repræsentationer og derfor bedre mentale modeller, er deres præstation af problemstillinger mere retvisende og detaljereret end nybegynderne.

Ud fra denne viden konkluderer Proctor and Zandt [2008][s. 336], at med praksis erfaring opnår en person domænespecifik viden og færdigheder, som kan bringes i anvendelse på den aktuelle opgave. Det er denne viden, der definerer en ekspert. Den domænespecifikke viden, som eksperter besidder, og hvordan den viden er organiseret, lader dem opfatte, huske og løse problemer bedre på dette område, end ikke-eksperter kan. Ekspertadfærd kan karakteriseres som færdighedsbaseret.

Santhanam and Wiedenbeck [1993][s. 226 - 227] fokusere sin artikel på de brugere, der er imellem nybegyndere og eksperter, som de kalder *discretionary users*. Denne type brugere bliver defineret af [Santhanam and Wiedenbeck, 1993] som værende brugere med betydelig erfaring og eksponering for systemet, men ikke har den bredde og dybde som ekspert brugeren har. I modsætning til nogle nybegyndere, der bruger systemet intensivt og med tiden bliver eksperter, nøjes denne type brugere med en stabil ydeevne ved at bruge systemet på en begrænset måde. Denne type brugere benytter kun et begrænset antal af kerne funktionerne i systemet for at opnå deres mål med programmet. Derfor vil et kompleks system med mange funktioner ikke være brugbar for denne del af målgruppen. Dette er grundet, at deres forståelse for et system er på et overordnet plan, hvor de har kendskab til at løse specifikke problemer, men de har ofte ikke nok forståelse omkring systemets struktur til at udføre nye opgaver. Grundet dette bør informationer omkring systemet blive præsenteret på deres fortrukne procedureform og ikke på et højere abstrakt plan. Dette uddyber Santhanam and Wiedenbeck [1993][s. 227] med at simple opgaver såsom kommandokald behøver denne type brugere kun en liste med forskellige kommandoer, hvor menuen eller listen bør være handlingsbaseret, såsom det at slette tekst eller sende tekst. For opgaver som ikke høre under rutineopgaver, har denne type brugere et behov for en mere guidede tilgang til læring, hvor de skal kunne se resultatet af deres handlinger hurtigt, da denne type brugere har et højt behov for at verficere deres handlinger.

2.3 Design for eksperter og nybegyndere

Dette afsnit har til formål, at undersøge hvordan man skal designe og strukturere dokumentation, der både har nybegyndere og eksperter i fokus.



Figur 2.2. Figuren viser indlæringskurven for to hypotetiske systemer, hvor det ene system kun fokusere på nybegyndere og det andet system kun fokuserer på eksperterne. Henad x-aksen er tiden repræsenteret og opad y-aksen er brugsfærdighed og effektivitet for brugeren. Taget fra Nielsen [1993][s. 26]

Som det ses på figur 2.2, vil et system ifølge Nielsen [1993][s. 28], der kun fokusere på nybegyndere være nemmere at lære, men mindre effektivt at bruge. Det modsatte er gældende for et system, der fokusere udelukkende på eksperterne. Dette understøtter Shneiderman [2004][s. 37], da han forklarer, at nybegyndere helst skal have et begrænset antal mulige handlinger, men efterhånden som brugernes erfaring øges, stiger deres ønske om øget funktionalitet og ydeevne også.

Man kan dog, ifølge Nielsen [1993][s. 41], designe et system, således at begge typers behov bliver dækket. Dette kan gøres ved at designe en brugergrænseflade med flere interaktionsmuligheder. Det vil skulle gøres på en sådan måde, at en bruger først lære systemet at kende via en interaktionsstil. Når brugeren dernæst bliver mere vant til at benytte systemet, kan denne ændre sit brug af systemet til en mere effektiv interaktionsstil. Shneiderman [2004][s. 37] foreslår, at man kan tilgodese både erfarne og nybegyndere i samme system ved at designe med en niveaustruktureret tilgang. Via denne tilgang kan nybegynderen benytte sig af de højere niveauer i strukturen, og jo mere erfaren brugeren bliver, kan de benytte de lavere dele af strukturen, som vil indeholde information og funktioner af højere kompleksitet.

En måde at implementere den niveaustruktureret tilgang er ifølge Nielsen [1993][s. 120 - 121], at benytte *Less is more princippet*. Med dette princip menes der f.eks. at designe en side med kun de mest essentielle informationer og henvise mindre vigtige informationer til sekundære sider. Det er derfor bedre, at formidle essentielle oplysninger på sin startskærm, da det kan tjene som feedback til brugeren om det er et relevant program, som brugeren er ved at benytte. Nielsen [1993][s. 121] forklarer, at en almindelig fejl, der bliver fortaget i udviklingen af brugergrænseflader er, at tilføje mange muligheder og stier til at løse en bestemt opgave. Dette er dog en fejl, da det at tilføje unødvendige stier og funktioner til et domæne, gør dokumentationen og domænet større og mere kompleks.

Fælles for alle typer brugere er ifølge Bhatti et al. [2021][s. 83 - 99], at kode eksempler også kaldt Working/Worked examples er essentielle for brugere af dokumentation. Grunden til dette forklarer Bhatti et al. [2021] er, at brugerne generelt ønsker i den indledende fase, at

skimte dokumentationen igennem, for dernæst at finde et kodeeksempel med det formål at forstå biblioteket. Dernæst ønsker brugeren ofte, at kopiere og indsætte eksemplet direkte i deres kildekode for at se, hvordan det virker. Bhatti et al. [2021][s. 83 - 99] forklarer, at generelt indeholder dokumentation to typer kodeeksempler: eksekverbar og forklarende eksempler. Den eksekverbare kode kan kompileres direkte ved at brugeren kan kopiere og indsætte koden. Forklarende kode forventes ikke at kunne kompileres direkte, da det normalt er et output, som en bruger kan lære af eller sammenligne med deres egen. Bhatti et al. [2021] forklarer at et godt kodeeksempel bør være forklaret i form af tekst. Teksten bør beskrive konteksten og forklare eventuelle forudsætninger for at køre en kodeeksempel, som f.eks. installation af specifikke biblioteker, samt begrænsninger som for eksempel hvis koden kun virker med visse versioner af et programmeringssprog.

2.4 Faktorer der influerer i valget af softwarebiblioteker

Larios Vargas et al. [2020] har i deres forskningsartikel undersøgt, hvilke faktorer der influerer udviklere i valget af softwarebiblioteker. Dette undersøgte de ved at udføre *cross-sectional* interviews med 16 udviklere fra 11 forskellige virksomheder, der alle havde mellem 4 til 15 års erfaring med programmering. Dernæst analyserede de resultaterne og for at få valideret deres resultater fra interviewene, blev der udarbejdet et spørgeskema, som blev besvaret af 115 respondenter.

Larios Vargas et al. [2020][s. 8] skriver, at selvom softwareindustrien betror sig meget på at genbruge tredjepart biblioteker, er valget af disse en krævende opgave og den vil i fremtiden kun blive mere krævende, da kompleksiteten og antallet af biblioteker vokser.

Larios Vargas et al. [2020][s. 9] udleder i deres forksningsartikel 25 faktorer, der influere i valget af trejdepartsbiblioteker. I dette afsnit vil der kun blive fremhævet de relevante resultater fra undersøgelsen ift. projektets formål. Faktorerne er inddelt i tre forskellige kategorier; Tekniske faktorer, økonomiske faktorer og menneskelige faktorer. Af faktorer der influere i kategorien tekniske faktorer, er der blandt andet, hvor passende biblioteket er ift. det ønskede formål. Larios Vargas et al. [2020][s. 5] forklarer, at udviklere er positivt stemte overfor biblioteker, som indeholder den ønskede funktionalitet med mindst mulig behov for brug af kode, der skal inkluderes i kildekoden. Dette kan f.eks. være i forhold til initialisering og konfiguration, som ønskes at blive holdt på et minimalt niveau.
En anden faktor, der er væsentlig for udviklere er kvalitetsrelaterede faktorer, hvor Larios Vargas et al. [2020][s. 5] blandt andet fremhæver dokumentation, usability og sikkerhed som essentielle faktorer. Ift. dokumentationen vurderede 51 % af deltagerne, at dette har stor indflydelse på deres valg af biblioteker, da god dokumentation er et tegn på god kvalitet af biblioteket. Her forklarede otte af de interviewede, at de som regel undersøger dokumentationen for gode eksempler og tutorials, der forklarer, hvordan man skal bruge biblioteket, inden de foretager deres valg. Larios Vargas et al. [2020][s. 10] konkludere også, at udviklere ofte vælger biblioteker fra med dårlig dokumentation, hvor der f.eks. er ingen til lav vedligeholdelsegrad og/eller dårlig ry i udvikler miljøer.

En anden kvalitetsrelaterede faktorer, der nævnes af Larios Vargas et al. [2020][s. 10] som værende essentiel for 42% af udviklerne er usability. Her nævner en af de interviewede, at hvis systemet er meget kompleks at bruge og kræver meget overvejelse, har denne person

en tendens til at vælge et andet bibliotek.

Af menneskelige faktorer der spiller ind ift. valg af biblioteker fremhæver Larios Vargas et al. [2020][s. 6], at udviklere vægter individuelle erfaring med biblioteket højt. Ydermere er erfaringer fra udviklermiljøet omkring biblioteket, de forskellige interessenter i projektet, og karakteristika ved den virksomhed/organisation, hvor biblioteket er udviklet essentielle i valget af biblioteker.

Af økonomiske faktorer, bliver risiko fremhævet af Larios Vargas et al. [2020][s. 8], da udviklere vurderer den mulige risiko som et bibliotek kan indeholde for et software system i valget af et biblioteket. Det nævnes af en af de interviewede, at i risikovurderingen indgår den estimeret tid på at integrerer et bibliotek i sin kildekode og hvor lang tid vil biblioteket være tilgængelig i fremtiden.

2.5 Problemformulering

Den gennemgået teori leder op til følgende problemformulering:

> "*Hvordan kan brugerdokumentation udvikles, så det kan benyttes på tværs af programmeringssprog og af personer med forskellige programmeringsevner?*"

Med udgangspunkt i denne problemformulering er der udarbejdet følgende forskningsspørgsmål:

- Forskningsspørgsmål 1: Hvordan anvender udviklere med forskellige programmeringsevner dokumentationen til tredjeparts biblioteker?
- Forskningsspørgsmål 2: Hvilke udfordringer opleves der i udviklingen af dokumentationen og i brugen af den?
- Forskningsspørgsmål 3: Hvilke ønsker har udviklere med forskellige programmeringsevner til indhold i dokumentationen for trejdeparts biblioteker?
- Forskningsspørgsmål 4: Hvilke designelementer, samt struktur skal dokumentationen af tredjeparts biblioteker have for, at tilgodese både erfarne og uerfarne brugers arbejdsmønstre?

2.6 Afgrænsning

Projektet vil afgrænse sig fra at udvikle og undersøge dokumentation i koden, og derfor vil projektet kun fokusere på den understøttede brugerdokumentation. Projektet vil kun have fokus på simple hjælper-biblioteker, som har enkle funktioner til rådighed.

Brugerundersøgelser 3

Dette kapitel har til formål at beskrive hvorledes to brugerundersøgelser er blevet udført. Disse har haft til formål, at undersøge forventninger til indhold af dokumentation fra både erfarne og uerfarne brugers perspektiv. Brugerundersøgelserne har ligeledes til formål, at undersøge hvilke forventninger brugerne har til design og struktur af dokumentationen til tredjeparts biblioteker.

Ud fra denne viden, vil der til sidst i kapitlet blive opstillet fokuspunkter til videre design. For at kunne opnå formålene med brugerundersøgelserne, er der blevet udført tre semistrukturet interviews, samt udsendt et online spørgeskema. Valget på disse to metoder er baseret på den gennemgået teori i afsnit 2.1.

3.1 Semistruktureret interviews

Følgende afsnit har til formål at gennemgå fremgangsmåden og resultaterne fra interviewsene. Interviewsene havde til formål at indsamle informationer omkring udviklingen af dokumentation, hvordan dokumentationen bliver brugt, samt de problemstillinger, der opstår i brugen af disse.

Rådata i form af samtykkeerklæringer, optagelser og transskribering fra interviewsene, kan findes i bilag A.

3.1.1 Introduktion

Ifølge Brinkmann and Tanggaard [2015][s. 34 - 37] er det nødvendigt, at man som interviewer har perspektiveret emnet som interviewet skal omhandle bredt forinden udførelsen af interviewet. Dette kan f.eks. gøres ved at læse om lignede studier og/eller undersøge medier for hvad de skriver om emnet. Denne forhåndsviden skal være med til at gøre det muligt for intervieweren at stille uddybende spørgsmål undervejs i interviewet. Brinkmann and Tanggaard [2015] forklarer ydermere, at endnu en fordel ved at have viden omkring emnet inden interviewet påbegyndes er, at dagsorden netop bliver sat af intervieweren og denne dagsorden bør være teoritisk begrundet. Dog understreger Brinkmann and Tanggaard [2015], at det er vigtigt at intervieweren går ind i samtalen med et åbent sind ift. hvad den interviewede siger.
Braun and Clarke [2006][s. 11] forklarer, at nogle vil argumentere for at man vil indsnævre sit analytiske synsfelt ved at have forhåndsviden. Dette kan derfor have den konsekvens, at man fokusere på bestemte dele af datasættet på bekostning af noget andet. Ifølge Braun and Clarke [2006][s. 11] vil andre dog argumenterer for, at det at have kendskab til litteraturen kan forbedre ens analyse, da man bedre opdager visse tendenser i datasættet. Ud fra

projektets formål er det blevet vurderet, at det er en fordel at have kendskab til emnet før interviewet, grundet det komplekse tema som projektet omhandler. Ud fra den gennemgået teori i kapitel 2, vurderes det ydermere, at der er skabt grundlag for en grundliggende viden omkring dokumentationen.

For at strukturere de tre interviews bedst muligt, er der blevet udarbejdet en interviewguide, som har til formål at skabe rammerne for interviewene og sikre at alle relevante emner bliver diskuteret under interviewet. Når spørgsmålene fra interviewguiden, samt eventuelle opfølgningsspørgsmål er gennemgået, afsluttes interviewet.

I appendiks A, kan den udarbejdet interviewguide findes.

Fremgangsmåde

Det vælges at udføre alle interviewes som semistrukturerede, da denne interview form giver mulighed for at stille uddybende spørgsmål til relevante emner, som kan opstå under interviewet og derfor ikke er planlagt på forhånd [Brinkmann and Tanggaard, 2015][s. 34 - 37]. En ulempe ved semistrukturede interviews er dog, at den kan begrænse muligheden for at interviewet kommer til at omhandle relevante emner, som der i interviewet ikke ledes op til. Det vurderes dog, at denne interview form er den mest fordelagtige med formålet taget i betragtning.

Ifølge Brinkmann and Tanggaard [2015] kan det ofte have en negativ effekt på flowet af ens interview, hvis intervieweren skal tage noter undervejs. Dette er grundet, at intervieweren allokerer mentale ressourcer på at nedskrive det sagte, hvilket gør at intervieweren vil have sværere ved at at være en del af samtalen, og stille uddybende spørgsmål under interviewet. Derfor vælges det at lydoptage interviewsene, som vil foregå via en telefon og en computer.

Respondenter

Interviewsene er blevet afholdt med tre udviklere individuelt.

Alle tre interviews er blevet afholdt i et mødelokale ved Seluxit. Mødelokalet gav mulighed for at afholde interviewet et sted, hvor der var ro og minimal forstyrrelse fra udefrakommende faktorer, og samtidig indeholdt mødelokalet whiteboards i tilfælde af noget skulle forklares grafisk.

Den ene respondent hedder Andreas og har været udviklingschef ved Seluxit i de sidste tre til fem år. Andreas programmere i JavaScript, TypeScript, C og Ruby, hvor C er det primære programmeringssprog. Den anden respondent hedder Karsten og er uddannet indenfor Computer Vision på Aalborg Universitet. Sidenhen har Karsten arbejdet 11 år ved CSR/Samsung og de efterfølgende 7 år ved Seluxit. Karsten programmere primært i C, men kan også programmere i Python og JavaScript. Den tredje respondent hedder Michael og han har arbejdet ved Seluxit siden 2016 og arbejder primært i Python, men kan dog også programmere i C og C++.

Det er valgt, at udføre interviews med netop disse udviklere for at få en bred indsigt i, hvilke holdninger og erfaringer der haves på tværs af forskellige programmeringssprog. Som beskrevet i afsnit 2.1, bør de interviewede udvælges således de er repræsentative for hele

populationen. Det har dog ikke ikke været muligt at afholde interviews med udviklere, der kan kategoriseres som uerfarne på dette stadie af projektet.

Analyse

Efter interviewende var overstået, blev alle interviews transskriberet, hvilket skete i programmet NVivo[1]. Transskriberingen foregik ved at gennemhøre lydoptagelserne op til flere gange. Ifølge Brinkmann and Tanggaard [2015] er det en fordel, at transskriberingen er udført af den som også har afholdt interviewet. Dette er grundet, at intervieweren vil kunne huske sociale og emotionelle relationer fra interviewet, som en person, der ikke deltog, nødvendigvis ikke ville kunne opfange fra lydoptagelsen. Lydfiler, samt transskriberingen af interviewene findes i bilag A i mappen "Interview - Udviklere ved Seluxit".

Resultaterne for interviewene er blevet analyseret via en tematisk analyse, der har taget inspiration fra Braun and Clarke [2006][s. 87 - 93] step-by-step guide, som indeholder seks steps. Det første step i denne guide er, at familisere sig med datasættet, hvilket skete både under selve interviewene, men også under transskriberingen af interviewene. Det andet step i analysen, var at udarbejde indledende *codes* fra datatsættet. I dette step af analysen, er der primært blevet brugt en induktiv tilgang til datasættet, da temaerne først er udledt efter inddeling af *codes* [Braun and Clarke, 2006][s. 83 - 84].
Under udarbejdelsen af *codes*, er der primært blevet ledt efter semantiske træk ved datasættet, da det ikke forsøges at udlede mere end det sagte. I step tre, fire og fem undersøges der for fælles temaer, hvor disse også navngives. I disse steps, er der også blevet frasorteret ikke relevante emner. Step seks er selve rapporteringsdelen af resultaterne, hvilket vil ske i det efterfølgende afsnit.

Analysen er blevet udarbejdet med en essentialistisk/realistisk tilgang, da der antages et simpelt, ensrettet forhold mellem mening og erfaring og sprog [Braun and Clarke, 2006][s. 85]. Temaer og *codes* er blevet udledt på baggrund af en subjektiv vurdering, hvor det er blevet vurderet om det sagte har været relevant i forhold til de overordnede forskningsspørgsmål.

3.1.2 Resultater

Dette afsnit har til formål at beskrive resultaterne fra analysen.

Efter at have fjernet alle *codes* som ikke er vurderet relevante ift. formålet med interviewene, blev der fundet 227 *codes*. Disse *codes* er efterfølgende blevet i opdelt i tre hovedgrupper; Udvikling af dokumentation, Brug af dokumentation og Tredjeparts bibliotek.

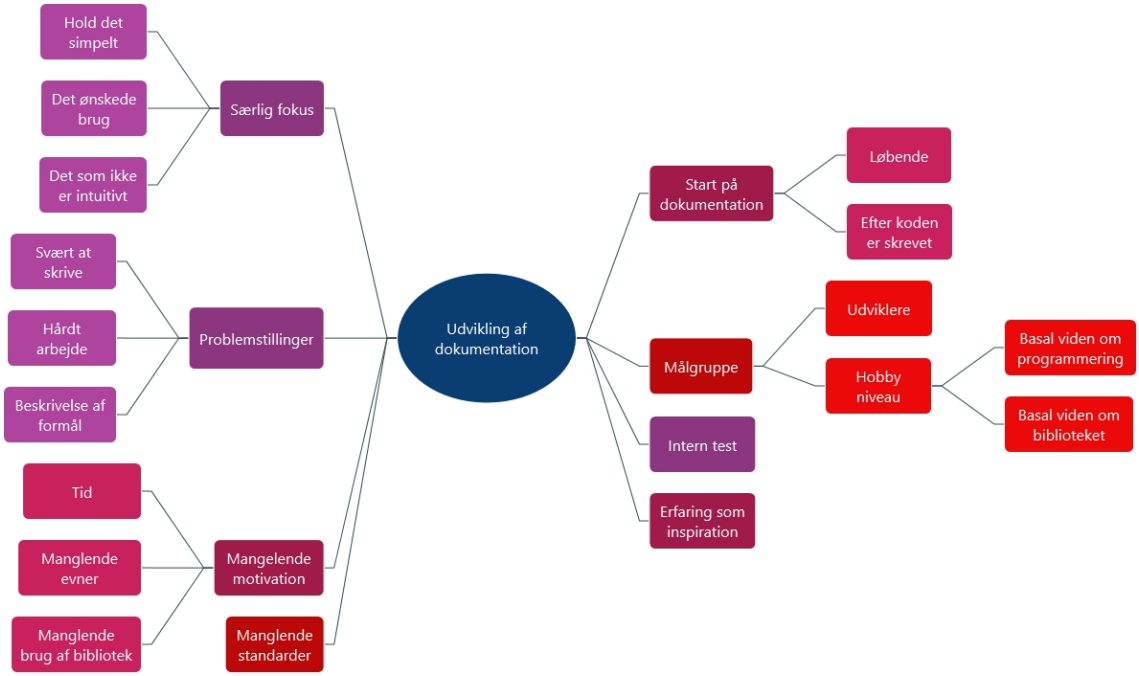
Fælles for alle tre affinity diagrammer i dette afsnit er, at den midterste blå boble er hovedgruppen, hvor hvert led ud i diagrammet er en undergruppe til den førværende gruppe. Se f.eks. figur 3.1.

[1] `https://www.alfasoft.com/dk/produkter/statistik/nvivo.html`

Udvikling af dokumentation

En af de større emner under de tre interviews var selve udviklingen af dokumentationen set fra udviklerens perspektiv. Samtalen blev præget i den retning af de forudbestemte spørgsmål fra interviewguiden (se appendiks A), men der opstod også nye emner, som blev fundet relevante i projektets øjemed.

Hovedgruppen "Udvikling af dokumentation", som ses på figur 3.1, er samlinger af *codes*, hvor respondenterne har udtalt sig om deres tanker og erfaringer med det at udvikle dokumentation til et trejdepart bibliotek. Denne hovedgruppe er inddelt i otte undergrupper og har i alt 62 *codes*.



Figur 3.1. Affinity diagram over gruppen; Udvikling af dokumentation.

Særlig fokus

En af undergrupperne er "Særlig fokus", som er omtalt af alle tre interviewede og har i alt 14 *codes*. Denne undergruppe er samlinger af *codes*, hvor respondenterne har udtalt sig om hvilke fokuspunkter de har haft, når de selv har udviklet dokumentationen.

En af undergrupperne til "Særlig fokus" er "Hold det simpelt", som omhandler vigtigheden i at holde dokumentationen på et simpelt niveau. Andreas udtaler blandt andet, at " man skal skære ned på, hvad der skal dokumenteres" og Michael udtaler "(..) der skal ikke være alt muligt unødigt ting, de skal tages stilling til".

I gruppen "Det ønskede brug" er det kun Michael, der er repræsenteret med en *code*. Her forklarer han, at han forsøger, at fokusere indholdet af dokumentation til, hvad han ønsker brugeren skal benytte i biblioteket.

Et emne, som har været i særlig fokus for to af respondenterne er undergruppen "Det som ikke er intuitivt", hvor der er udledt fire *codes*. Her beskriver Andreas blandt andet, at han har særlig fokus på de emner, der ikke er intuitive for brugeren i sin udvikling af

dokumentation. Karsten beskriver dette ved at udtale "Det jeg fremhæver er det jeg tror folk vil have sværest ved at forstå.".

Problemstillinger
En anden undergruppe i "Udvikling af dokumentation" er "Problemstillinger". Denne undergruppes indhold omhandler problemstillinger, som de tre interviewede mener, at de har ift. udvikling af dokumentationen. Denne undergruppe har i alt 12 *codes*, hvor kun to af respondenterne er repræsenteret i denne gruppe.

I undergruppen "Svært at skrive" nævner både Andreas og Karsten, at udarbejdelsen af dokumentationen finder de svært. Andreas forklarer dette via følgende; "(..) alt det der udenom som er det, som man selv, når man læser funktionen vil have står der, men selv skulle skrive, det er det svære.". Karsten beskriver lignede problemstilling " (..) det er svært at finde ud af, hvad for nogle ting, der er vigtige at beskrive og beskrive det på en måde, som folk når de ikke ved hvad det er forstår det".

Udover at dokumentationen er svært at skrive, forklarer Andreas i undergruppen "Hårdt arbejde", at det at skrive dokumentationen er hårdt arbejde, da det tit bliver indforstået, det der bliver skrevet. Han udtaler følgende: "(..) man skal altså skrive forståelig tekst fordi det bliver tit indforstået. Det bliver for specifikt og for teknisk fordi du skal beskrive det på en måde, så folk der ikke ved hvad det er kan læse det".

Andreas uddyber dette i gruppen "Beskrivelse af formål", som han også ser som værende svært at få beskrevet i dokumentationen ved at udtale: "det er nemt nok at dokumentere en funktion på hvad den gør, men det er svære at dokumentere hvorfor du skal bruge den". Denne gruppe har to *codes*, hvor begge fra Andreas.

Manglende motivation
Denne undergruppe er en samling af de udtaleser fra respondenterne, som omhandler, hvorfor respondenterne har en manglende motivation for at udvikle dokumentation. Denne undergruppe er repræsenteret af to respondenterne med i alt ni *codes*. Som det ses på figur 3.1, har undergruppen selv tre undergrupper i tredje led og de tre undergrupper er "Tid", "Manglende evner" og "Manglende brug af bibliotek".

Undergruppen "Tid" omhandler, at de interviewede mener, at der skal bruges en stor mængde tid på at udvikle ordentlig dokumentation. Til dette emne udtaler Andreas følgende: "man ved at det tager lang tid at skrive det ordentligt og derfor gør man det ikke". Karsten beskriver til samme emne "(..) man kan brænde en masse tid af på dårlig dokumentation".

Ydermere forklarer de samme to personer i gruppen "Manglende evner", at det der ligeledes demotivere dem i udarbejdelsen af dokumentation er manglende evner indenfor det. Her udtaler Andreas følgende "Fordi jeg føler også at jeg har næsten ikke lyst til at skrive noget, der ikke er godt nok fordi det skal slettes og skrives om".

Manglende standarder
I denne undergruppe er der samlet de udtaleser, der omhandler manglende standarder indenfor dokumentation. Denne gruppe indeholder fem *codes* fra to af respondenterne. Til spørgsmålet om der findes standardisering i hvorledes dokumentation skal udformes, svarer

Andreas "(..) ikke i *libraries* på den måde, ikke hvad jeg ved af. Så er det altid indenfor et specifikt domæne, der er nogle standarder, men generelt er der ikke noget."

Start på dokumentation
Denne undergruppe omhandler, hvornår respondenterne starter på dokumentationen af deres biblioteker og der er i alt fem *codes* og alle respondenterne er repræsenteret i denne gruppe.

Andreas er begyndt at skrive dokumentationen løbende, da han udtaler, i undergruppen "Løbende" følgende; "(..) først når jeg har skrevet den lille klump dokumentation til den lille funktion jeg har lavet er den færdig.". Dette er på trods af, at han tidligere har startet på dokumentationen efter biblioteket var færdigt. Michael udtaler, at han laver *inline*-dokumentation løbende, men ellers lader han andre om, at udarbejde dokumentationen. Karsten udtaler i gruppen "Efter koden er skrevet", at han først starter på udarbejdelsen af dokumentationen, når han er færdig med selve koden.

Målgruppe
Denne undergruppe omhandler, hvilken målgruppe respondenterne har haft i fokus i deres udvikling af dokumentationen. Undergruppen har 22 *codes* i alt og alle interviewede er repræsenteret i denne gruppe.

Der er en general enighed blandt respondenterne om, at dokumentationen til biblioteker, der skal bruges til ekstern brug, skal rettes imod folk på hobbyniveau. Med hobbyniveau menes der at de har en basal viden om både programmering generelt og biblioteket selv. Til dette forklarer Michael blandt andet "Jeg forventer de ved, det basale med hvad det er til for f.eks. JSON RPC, der forventer jeg, at de ved hvad JSON RPC er". Andreas forklarer til samme emne: "(..) en forståelse af dit *enviroment* og du ved hvad Wappsto er og hvad Wapps er og du ved hvad JavaScript *libraries* er".

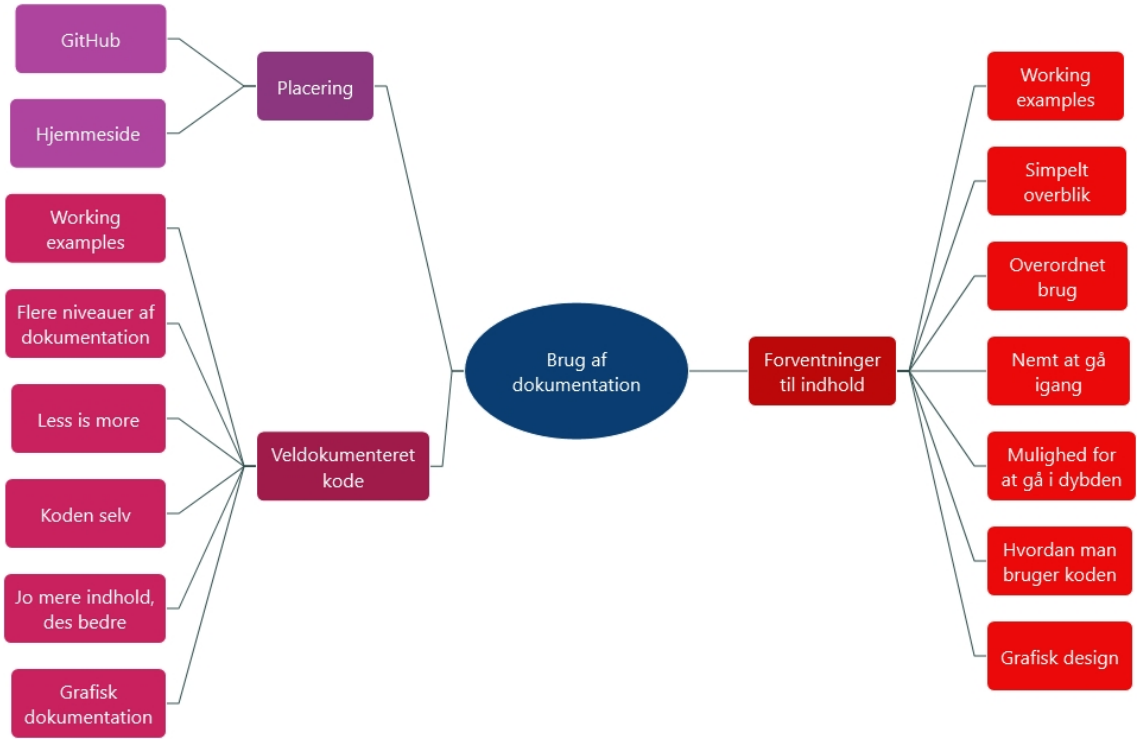
Dog forklarer Karsten i undergruppen "Udviklere", at han også har haft fokus på folk på et højere ekspertiseniveau. Dette er grundet, at han har skrevet dokumentation, som sommetider har været henvendt til kunder, som har personel på samme programmeringsniveau som Karsten selv.

Intern test
Intern test omhandler, hvordan respondenterne tester sin dokumentation. Dog fik Karsten kun dette spørgsmål og er derfor den eneste repræsenteret i denne undergruppe med to *codes*. Her forklarer Karsten til spørgsmålet om hvordan han tester forståelsen af det udviklet dokumentation, at det er ved at spørge kollegaer.

Erfaring som inspiration
Denne undergruppe omhandler, hvad respondenterne bruger som inspirationkilde til deres dokumentation. Undergruppen er repræsenteret af alle tre interviewede med otte *codes* i alt. Her var der en general enighed om, at det er erfaringsbaseret inspiration, der bruges. Her forklarer Karsten blandt andet "Der er et hav af forskellige måder at gøre det på og man prøver at huske de steder, hvor man har fundet noget der var godt" og Michael forklarer "(..) med hjemmesider, så er man nød til at tage den erfaring man selv har fået igennem tiden og det er nok lidt groft at sige, men dem man husker, det er dem der forfærdelige".

Brug af dokumentation

Denne hovedgruppe omhandler respondenternes erfaring og holdninger indenfor brug af dokumentationen.



Figur 3.2. Affinity diagram over gruppen; Brug af dokumentation.

Hovedgruppen har tre undergrupper kaldet; "Placering", "Veldokumenteret kode", "Forventninger til indhold". Hovedgruppen har i alt 110 *codes* og alle tre interviewede personer er repræsenteret. Fordelingen af undergrupper i andet og tredje led i hovedgruppen, kan ses på figur 3.2.

Placering

Denne undergruppe omhandler, hvor respondenterne vil forvente og hvor de vil fortrække at finde dokumentationen for et bibliotek. Der er i alt 20 *codes* i denne undergruppe og alle tre personer, der blev interviewet er repræsenteret.

En general holdning blandt respondenterne til placering af dokumentationen var, at det er forventede at dokumentationen skal kunne findes enten på udvikler platformen GitHub[2] eller på en dedikeret hjemmeside. I forhold til den dedikeret hjemmeside forklarer Andreas følgende "altså hvis det er et stort projekt, så har de en hjemmeside om deres *library* ellers er det bare en Readme fil på GitHub.". Karsten forklarer lignende "Nogle få, enkle er så store at de har deres hjemmesider, hvor de går ind og *maintainer* det". Angående GitHub forklarer Karsten følgende "mange af dem, selvom de er store og har deres egen hjemmeside, så lægger al koden stadig på GitHub og så er der referencer frem og tilbage".

[2]https://github.com/

Veldokumenteret kode
Denne undergruppe omhandler, hvordan respondenterne vil definere veldokumenteret kode. Der er i alt 42 *codes* og alle tre interviewede er repræsenteret i denne undergruppe. Som det ses på figur 3.2 blev der fundet seks generelle emner, som respondenterne har nævnt, hvor "Working examples" er en af dem. Her udtaler Michael f.eks. "jeg bare kommet til det punkt, hvor mere med *working examples* med kort beskrivelse. Det er det jeg kan lide".

Alle tre respondenter var enige om, at dokumentationen bør indeholde flere informationsniveauer. Med dette menes der, at det øverste niveau bør primært henvende sig til nybegynderen, hvor der er få og relativ simple informationer. Det næste niveau, bør så henvende sig mere til udviklere, som er bekendt med biblioteket, hvor udvikleren har mulighed for at tilgå avanceret informationer. Dette bliver f.eks. forklaret af Michaels kommentar "Der er to dybder af koden, ikke. Der er den du klikker på, når du først lærer det og så er den, når du kan det og vil gøre noget avanceret". Til det samme emne udtalte Karsten følgende "Den overordnet er når man går i gang (. . .) Når man kommer ind i det og siger hvorfor virker det her ikke, så skal man ind og grave i lige præcis den her og hvad sker der her".

Undergruppen "Less is more", som ses på figur 3.2, inkludere alle tre personers holdninger. Denne undergruppe omhandler udtalelser om, at holde dokumentation på et simpelt overskueligt niveau, hvor brugeren kun ser det nødvendige på siden. Karsten forklarer blandt andet følgende "Less is more, det bliver hurtigt et monster at lave sådan nogle ting, da man tit ender med mere kræver mere kræver mere". Michael forklarer til samme punkt følgende "(..) Man skal minimere mængden af den viden folk skal have proppet ind i hovedet. Jo mere du kan gøre det (..) jo større chance er der for at du får folk til at bruge det".

Andreas er som den eneste repræsenteret i undergruppen "Koden selv" på figur 3.2 med tre *codes*. Her udtaler Andreas: "(...) den bedste dokumentation, det er koden selv". Til samme emne forklarer han dog, at han ikke længere vil forvente at finde dokumentation i koden, men at det er bedre at ens kode er læselig, da det kan hjælpe ift. hvis dokumentationen ikke er opdateret.

Mængden af dokumentation kan ifølge respondenterne også have et kvalitetsaspekt. Alle tre interviewede nævner, at det føles mere gennemarbejdet, hvis der er meget dokumentation. Her siger Andreas blandt andet i undergruppen "Jo mere indhold, des bedre. Det er jo tit, at jo mere dokumentation der er, jo højere kvalitet er det, for det meste".

I undergruppen "Grafisk dokumentation" er kun Karsten repræsenteret med tre *codes*. I denne undergruppe forklarer Karsten, at det der kan repræsenteret grafisk, bør blive det.

Forventninger til indhold
Undergruppen "Forventning til indhold" indeholder holdninger og erfaringer til, hvad respondenterne vil forvente af indhold i dokumentationen til et trejdeparts bibliotek. I denne undergruppe er alle tre personer repræsenteret og der er i alt 48 *codes*.

Det kan ses på figur 3.2, at forventningerne til indholdet af dokumentationen er, at dokumentationen skal være nem at gå til, give et simpelt overblik over biblioteket og vise den overordnet brug af dokumentationen. Disse emner kan f.eks. eksemplificeres via Michaels kommentar "(...) det skal være nemt at komme i gang med ellers får man ikke

folk til det.".

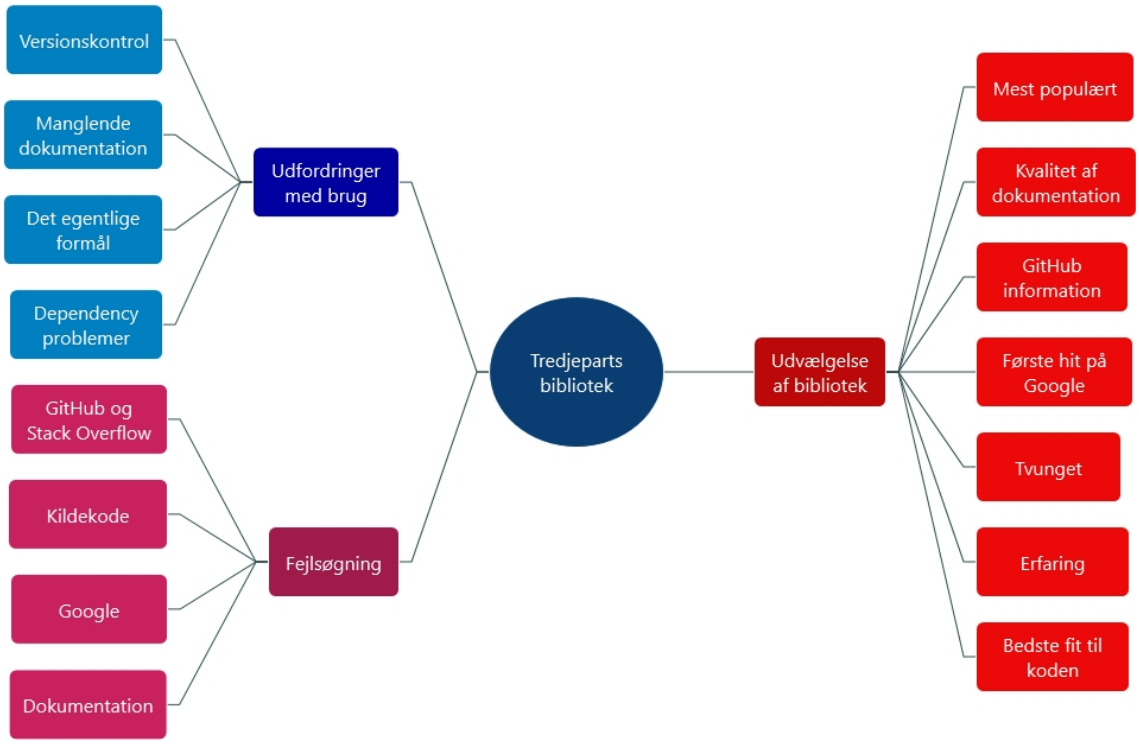
Alle respondenter forventede ydermere *working examples* og at det skal være muligt at dykke flere niveauer ned i koden. Her udtaler Michael blandt andet sig omkring det at gå i dybden med koden; "(...) helst nogle forskellige eksempler, og så kan vi gå ind og få den mere uddybende om de forskellige ting, hvis vi får lyst".

Både Andreas og Michael nævner også, at dokumentationen skal forklare hvordan man bruger koden. Her udtaler Andreas blandt andet, at "En ting jeg syntes der er irriterende, når jeg læser dokumentationen er, at de har en forventning om, at du ved det hele".

Alle tre interviewede er enige om at det grafiske design ift. flow af koden, hightlighting af kode og lignede bør være en del af dokumentationen. Her udtaler Karsten blandt andet, at "(...) gerne have det dokumenteret enten via et flowdiagram eller et *sequence chart* eller et eller andet grafisk".

Tredjeparts bibliotek

Denne hovedgruppe omhandler respondenternes erfaring og holdninger til forskellige emner indenfor brug af trejdeparts bibliotek. Denne undergruppe indeholder i alt 55 *codes* og alle tre respondenter er repræsenteret.



Figur 3.3. Affinity diagram over gruppen; Tredjeparts bibliotek.

Som det ses på figur 3.3, har denne hovedgruppe tre undergrupper, som hver har imellem fire til syv undergrupper yderligere i tredje led. De tre undergrupper i andet led hedder; "Udfordringer med brug", "Fejlsøgning" og "Udvælgelse af bibliotek".

Udfordringer med brug

I denne undergruppe forklarer respondenterne, hvilke udfordringer de generelt set, har med at bruge trejdeparts bibliotekerne.

En af undergrupperne er "Versionskontrol", som omhandler, det at bibliotekerne har forskellige versioner, kan skabe udfordringer for brugeren af biblioteket. Hertil forklarer Karsten "Hvis dokumentationen er gået ud af trit med nyeste version, så er det svært finde ud af, hvordan jeg skal bruge den".

En anden general udfordring med brug af biblioteker er undergruppen "Manglende dokumentation" som omhandler manglende eller misvedligeholdt dokumentation. Til dette fortæller Michael om en erfaring han gjorde sig ved at bruge et bestemt biblioteket: "(...) alt afhængigt af hvad filen hedder så skal de være opbygget på forskellige måder og det stod der heller ikke noget om".

Karsten forklarer, at han har oplevet, at beskrivelsen ikke altid stemmer overens med hvad tredjeparts biblioteket rent faktisk gør. Dette hører under undergruppen "Det egentlige formål". Karsten forklarer følgende "(...) med *open source libraries* er det ikke dem allesammen, der gør det de siger de gør og virker på den måde de siger de gør".

Dependency-problemer er en anden type udfordring, som Andreas bl.a. har oplevet. Hertil forklarer Andreas, at det er en ny type udfordring for *supply chain* angreb indenfor programmeringssprogene Python, Rust og JavaScript. Hertil bliver der forklaret, at dette skyldes at når man installere et biblioteket, så er dette som ofte afhængig af mange andre biblioteker. Hvis en af de andre biblioteker fejler eller ikke bliver opdateret, så rammer det også det biblioteket man bruger.

Fejlsøgning
Denne undergruppe omhandler hvad respondenterne gør, hvis der opstår fejl eller de sidder fast et sted ift. brug af et bibliotek. Der er en general enighed blandt respondenterne om, at man søger hjælp ved Stack Overflow[3], søger efter fejlen på Google eller lignede søgemaskiner eller kigger i selve koden. Michael forklarer ift. Stack Overflow følgende "hvis det ikke er noget der kommer frem på deres hjemmeside, så er Stack overflow typisk stedet man støder ind i" og Karsten forklarer følgende "Hvis det så ikke virker, jamen så må du ind i koden og finde ud af hvorfor det ikke virker."

Udvælgelse af bibliotek
Denne undergruppe omhandler, hvilke faktorer der gør sig gældende, når der skal udvælges et bibliotek. Her er alle tre respondenter enige om, at kvaliteten af dokumentationen er en vigtig faktorer..

En anden faktorer der har indflydelse i valget af bibliotekter er, hvilke biblioteker der er mest populære. Michael begrunder dette med følgende "Det der er mest brugt, det der er mest kendt, er jo typisk også det der er mindst fejl i".

De første hit på Google, har også en indflydelse og dette hænger sammen med hvor populært biblioteket er. Her forklarer Michael "De første par stykker skal i hvert fald være idiotisk opbygget før man begynder at vælge nogle der ikke er top tre i det du søger efter".

[3] `https://stackoverflow.com/`

Informationerne man får via GitHub, såsom problemer, opdateringer og lignede har også indflydelse i valget af biblioteket. Undergruppen "Tvunget" omhandler det, at man kan være tvunget til at bruge et bestemt bibliotek. Til dette nævnes der to forskellige grunde. Den ene grund forklarer Andreas er, at hvis man vil benytte et specifikt system som f.eks Couchbase[4], så er man tvunget til at bruge netop deres bibliotek. Michael forklarer, at en anden grund kan være, hvis der ikke er udviklet andet lignende og man derfor er nød til at bruge biblioteket.

I undergruppen "Erfaring" forklarer Andreas, at han vælger biblioteker blandt andet på baggrund af erfaringer fra tidligere brug. Her udtaler han "Hvis du har brugt det før, så er det nemmere igen og så tager man det samme *library* igen". I undergruppen "Bedste fit til koden" forklarer Karsten, at han udvælger et bibliotek baseret på, hvad der er det bedste fit til koden.

3.1.3 Diskussion

Følgende afsnit har til formål at diskutere de fundne resultater fra den tematiske analyse og årsagen til disse.

Alle tre respondenter, som blev interviewet er kendt af intervieweren. Dette kan have forårsaget en vis form for *pleaser*-effekt i interviewet, hvor respondenten forsøger at svare på spørgsmålene ift. hvad der antages intervieweren ønsker af resultater. Ydermere har alle tre interviewede haft kendskab til projektets formål, hvilket igen kan have forstærket *pleaser*-effekten [Brinkmann and Tanggaard, 2015][s. 36]. Dog kan man argumenter for, at grundet resultaterne er blevet sammenlignet på tværs af interviews og alle interviews er blevet holdt enkeltvis, vil ikke retvisende kommentarer være blevet frasorteret under analysen.

Denne rapport er udarbejdet af en person og det har derfor været den samme person som har fortaget interviewsene, udført transskriberingen og lavet analysen. Dette kan have haft den konsekvens, at visse *codes* og temaer er blevet udeladt og dette kan derved have påvirket reliabiliteten. Her kan man diskutere om, at det havde været en fordel at udføre analysen ved at samle en gruppe personer og i fællesskab finde temaer. På denne måde vil man måske kunne have fået et mere nuanceret indblik i datasættet og analysen vil ydermere blive et resultat af fleres holdninger i stedet for kun en persons.
Der har dog været stort fokus på at opretholde reliabiliteten ved at iterere over datasættet flere gange og forsøgt at holde samme fokus på alle dele af datasættet. Ydermere kan man diskutere om, det i visse aspekter af analysen, har været en fordel at analysen er udført af en enkeltperson. Dette er grundet, at samme person som fortager analysen også har udført interviewet og derved har haft mulighed for at genkalde ordlyden og konteksten emnerne er blevet sagt i.

Ifølge Guest et al. [2006] kan mætning af datasættet ske indenfor tolv interviews og allerede indenfor seks interviews kan de mest basale temaer opstå. Der er kun udført tre interviews i dette projekt, hvilket muliggøre at der ikke er opnået mætning i datasættet. En indikation på dette kan f.eks. være at der er enkle undergrupper der kun er repræsenteret af en eller to personer. Derfor kan man diskutere om der burde have været rekvireret flere respondenter

[4] `https://www.couchbase.com/`

for at få mættet datasættet. Dette er dog blevet fravalgt, grundet et lavt antal tilgængelige udviklere og den gennemgået teori i afsnit 2.1, hvor Bhatti et al. [2021][s. 12] anbefalede tre til fem personer. Det lave antal respondenter, der er blevet interviewet kan også have haft en negativ effekt på den eksterne validitet. Derfor kan der argumenteres for, at der bør laves en anden undersøgelse, der vil øge den eksterne validitet som f.eks. et spørgeskema.

En mulig begrænsning i interview metoden er, at hvis man kun benytter denne metode i analyse af målgruppen, så får man kun et indblik i hvad målgruppen siger de gør og ikke hvad de rent faktisk gør. Dette kan have medført, at de interviewede har en opfattelse af handlinger de fortager sig, som ikke er svarende til hvad de rent faktisk gør. Derfor kunne man have valgt at udføre et observationsstudie, for at få de usagte emner og handlinger med i analysen. Dette er dog blevet fravalgt grundet tidsøkonomiske årsager, samt et lavt antal tilgængelige forsøgspersoner.

Ud fra Braun and Clarke [2006][s. 81 - 83] forklaring af deduktiv og induktiv tilgang til analysen, er der primært blevet udført en induktiv tilgang, da temaer først er fundet efter inddeling af *codes*. Analysen kunne dog også have været udført med en deduktiv tilgang, hvor temaer defineres fra starten, og *codes* derefter lægges ind under disse. Dog vil den deduktive tilgang risikere at medføre, at *codes* der ikke indgår i de prædefinerede temaer kan gå tabt.

3.1.4 Opsummering af væsentlige resultater

Følgende afsnit har til formål at opsummere de mest væsentlige resultater fra interviewsene.

Som resultat af de tre interviews, gav respondenterne generelt udtryk for en manglende motivation for, at skulle udvikle dokumentation til deres biblioteker. Den mangelende motivation blev begrundet med, at det er en tidsmæssig krævende opgave at udarbejde tilfredsstillende dokumentation. Flere af respondenterne udtalte ydermere, at de havde svært ved formidle bibliotekets formål og brug til brugeren på hobbyniveau.

Respondenterne havde særlig fokus på at holde dokumentationen på et så simpelt niveau som muligt og kun beskrive det, som ikke virkede intuitivt for dem. Yderemere havde de fokus på at formidle formålet med biblioteket. Målgruppen de havde i fokus var generelt personer, der har et basalt kendskab til programmering og selve biblioteket.

Ift. brugen af dokumentation af trejdepartsbiblioteker, viste analysen at respondenterne generelt var enige om, at en veldokumenteret kode er grafisk dokumenteret med f.eks. *working examples*. Ydermere var der en general enighed om, at dokumentation bør indeholde flere niveauer. På denne måde kan den erfarne bruger udlede avanceret informationer omkring biblioteket, hvor nybegynderen kan få det nemme overblik. Der var uenighed blandt respondenterne omkring mængden af dokumentation, da nogle mente, at der kun skulle være det nødvendige og andre så kvantiteten som et kvalitetsstempel.

Af indhold i dokumentationen forventede respondenterne blandt andet, at det skulle være nemt at skabe sig et hurtigt overblik, at det skulle være nemt at gå i gang og at der var *working examples*.

Ved brugen af trejdepartsbiblioteker generelt oplevede respondenterne, at de ofte oplevede

fejl i dokumentationen ift. forskellige versioner af biblioteket. Yderemere oplevede de, at den dokumentationen ofte var mangelfuld. Ligeledes havde enkelte respondenter oplevet *dependency*-problemer.

Resultaterne viste, at respondenterne fejlsøger ved enten at søge på problemet på en søgemaskine som Google, kigger efter lignede problemstillinger på GitHub eller Stack Overflow eller undersøgte dokumentation for svar.

Når respondenterne skulle udvælge biblioteker, er det sket efter forskellige parametre som f.eks. hvilket der er mest populært, kvaliteten af dokumentationen, tidligere erfaring med biblioteket og hvordan biblioteket passer til koden.

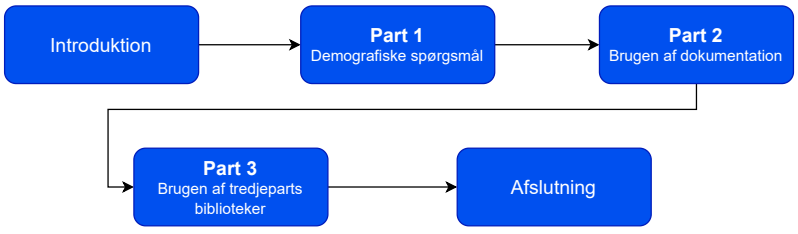
3.2 Spørgeskema

Da der på nuværende stadie kun er indsamlet kvalitativ data fra tre udviklere, ønskes det at indsamle viden fra en større del af målgruppen. Dette er for at sikre at resultaterne fra interviewsene er repræsentativ for populationen. I afsnit 2.1 blev det gennemgået, at spørgeskemaer netop opfylder dette formål og derfor vælges det at udsende et spørgeskema. Ydermere ønskes det, at undersøge om der er forskel i ønsker og behov for erfarne og uerfarne brugere.

Gennemgang af de udarbejdet spørgsmål, argumenter for disse og de overordnede resultater, samt rekvirering af respondenter, kan ses i appendiks B. Selve spørgeskemaet, samt den kode der er blevet brugt til grafer og udregninger, kan findes i bilag A i mappen "Spørgeskema".

3.2.1 Introduktion

Det vælges at udarbejde et spørgeskema, hvor spørgsmålene er blevet udarbejdet med inspiration fra resultaterne fra interviewsene, beskrevet i afsnit 3.1.2.



Figur 3.4. Figuren viser flowet for spørgeskemaet.

Som det ses på figur 3.4 er spørgeskemaet opdelt i fem dele. I første del får respondenten en introduktion til spørgeskemaet, dernæst i "Part 1" bliver respondenten stillet demografiske spørgsmål. I "Part 2" får respondenten spørgsmål til brugen af dokumentation og i "Part 3" omhandler spørgsmålene brugen af tredjeparts biblioteker. Til sidst takkes respondenten for deltagelse.

Pilottest

Spørgeskemaet blev udformet ved, at der først blev prioriteret hvilke emner, der var relevante ud fra formålet med spørgeskemaet. Ud fra disse emner, blev der udformet et første udkast af spørgeskemaet. Dernæst blev spørgeskemaet gennemgået med vejleder, hvor der var særlig fokus på strukturen af spørgeskemaet, samt relevansen af spørgsmålene. Efter spørgeskemaet var rettet til, blev spørgeskemaet gennemgået via to iterationer med udviklere, der er ansat ved Seluxit. Her var der særlig fokus på, at få formuleringer til at lyde fagligt korrekt ift. programmerings miljøet.

Respondenter

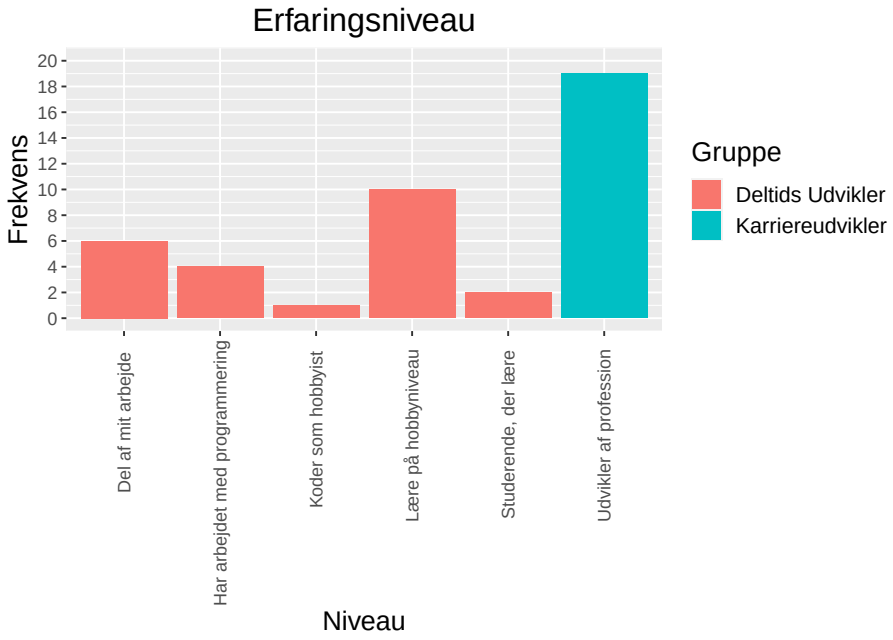
Spørgeskemaet var offentligt tilgængelig i perioden 23/3-2022 kl. 11:00 til 27/3-2022 kl. 12:00. Årsagen til denne tidsperiode er, at der ikke havde været aktivitet omkring spørgeskemaet i 24 timer på dette tidspunkt.

Spørgeskemaet blev besvaret af 41 respondenter, hvor otte af disse kun delvis gennemførte spørgeskemaet. Ud af de 41 respondenter, var der 35 mænd, 2 kvinder og fire der definere sig som noget andet. 35 af respondenterne var bosiddende i Danmark, 2 respondenter i Frankrig, 1 i Holland, 4 i USA og 2 respondenter afgav ikke konkret, hvor de boede.

3.2.2 Resultater

Dette afsnit har til formål, at undersøge om der er forskel på, hvad erfarne udviklere og uerfarne udvikler forventer i forbindelse med dokumentation af trejdeparts biblioteker.

Afsnittet vil kun opsummere væsentlige resultater. Den overordnet analyse af respondenternes samlede svar undersøges og beskrives i appendiks B.1.

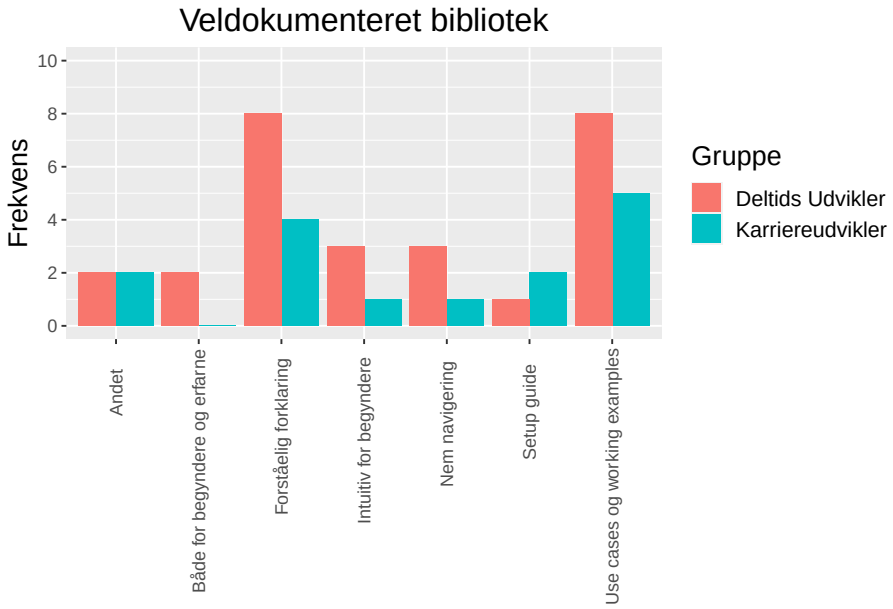


Figur 3.5. Figuren viser fordelingen af de to grupper "Deltids Udvikler" og "Karriereudvikler". Henad x-aksen er hvad respondenterne selv mener er mest signede for dem og opad y-aksen er det procentmæssige frekvens.

Fælles for graferne i dette afsnit er, at de respondenter der har svaret, at de er udviklere af profession er samlet under "Karriereudvikler", hvor resten af respondenterne er samlet under "Deltids Udvikler". Disse to grupper vil blive benævnt således igennem rapporten.

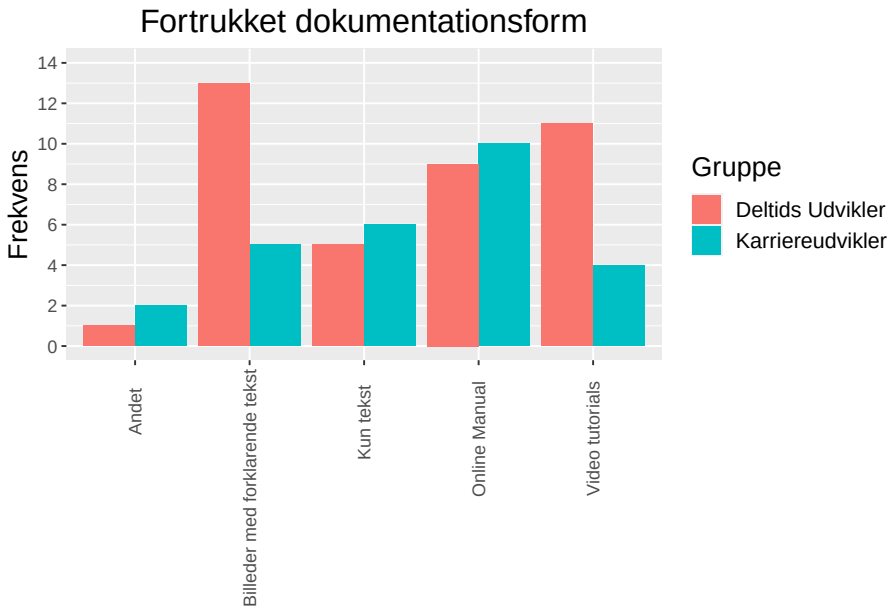
På figur 3.5 ses fordelingen af de to grupper. Der er i alt samlet 23 respondenter i gruppen "Deltids udvikler" og 19 respondenter i gruppen "Karriereudvikler".

Adskillelsen af grupperne er sket, da det er i projekts formål at undersøge hvilke forventninger uerfarne og erfarne brugere har til dokumentation af tredjeparts biblioteker. Opdelingen af "Deltids udvikler" og "Karriereudvikler" ift. demografiske spørgsmål, kan ses i appendiks B.2. I samme appendiks findes også resultater fra spørgsmål, som er vurderet mindre væsentlige for videreudviklingen af dokumentationen.



Figur 3.6. Figuren viser fordelingen af de to grupper "Deltids Udvikler" og "Karriereudvikler" holdning til et veldokumenteret bibliotek. Henad x-aksen er emner som respondenterne mener er vigtige for at et bibliotek er veldokumenteret. Opad y-aksen er det frekvensen repræsenteret.

På figur 3.6 ses det, at både gruppen Karriereudviklere og Deltids udviklere primært har nævnt *Use cases og working examples* som indhold, der bør være til stede i et veldokumenteret bibliotek. Det er blevet vurderet af fem karriererudviklere og otte respondenter fra "Deltids Udvikler" gruppen. "Forståelig forklaring" er også nævnt af otte respondenter af Deltids Udviklerne. "Forståelig forklaring" er nævnt af fire karriereudvikler, hvilket gør det til denne gruppes anden mest valgte emne. Emnet "Forståelig forklaring" indeholder svar omhandlende forklaring af formål, samt brug af funktioner og variabler.

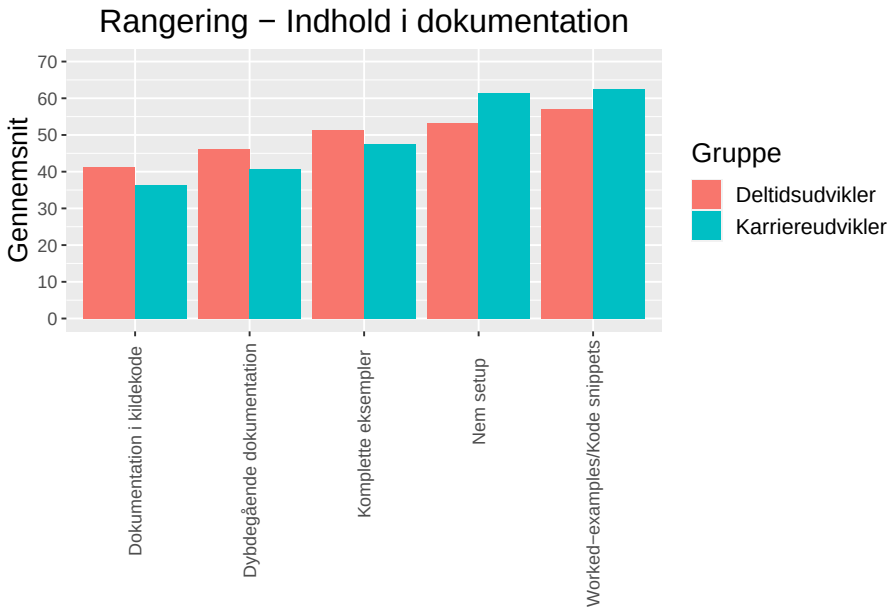


Figur 3.7. Figuren viser fordelingen af de to grupper "Deltids Udvikler" og "Karriereudvikler" ift. fortrukket dokumentationsform. Henad x-aksen er de valgte emner. Opad y-aksen er det frekvens repræsenteret.

På figur 3.7 ses det at karriereudvikler ønsker online manualler som dokumentationsform, hvilket gør det til den primære fortrukket dokumentationsform for denne gruppe. Sekundært fortrækker karriereudvikler kun tekst, hvilket er valgt af seks respondenterne.

Deltids Udviklere fortrækker primært billeder med forklarende tekst som dokumentationsform, hvilket ønskes af 13 af respondenterne. Sekundært bliver video tutorials fortrukket af 11 respondenter.

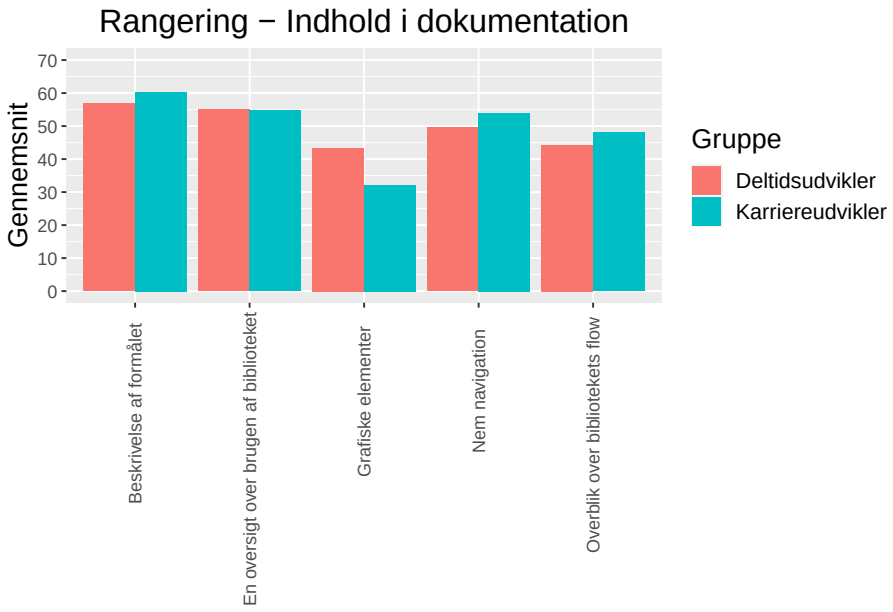
Respondenterne kunne vælge op til flere emner, hvilket gør, at de kan være repræsenteret i flere af emnerne i barplottet.



Figur 3.8. Figuren viser hvordan de to grupper "Deltids Udvikler" og "Karriereudvikler" har rangeret ift. indhold i dokumentationen. Henad x-aksen er de valgte emner og opad y-aksen er gennemsnittet. For at se de eksakte værdier se tabel B.6 og B.7.

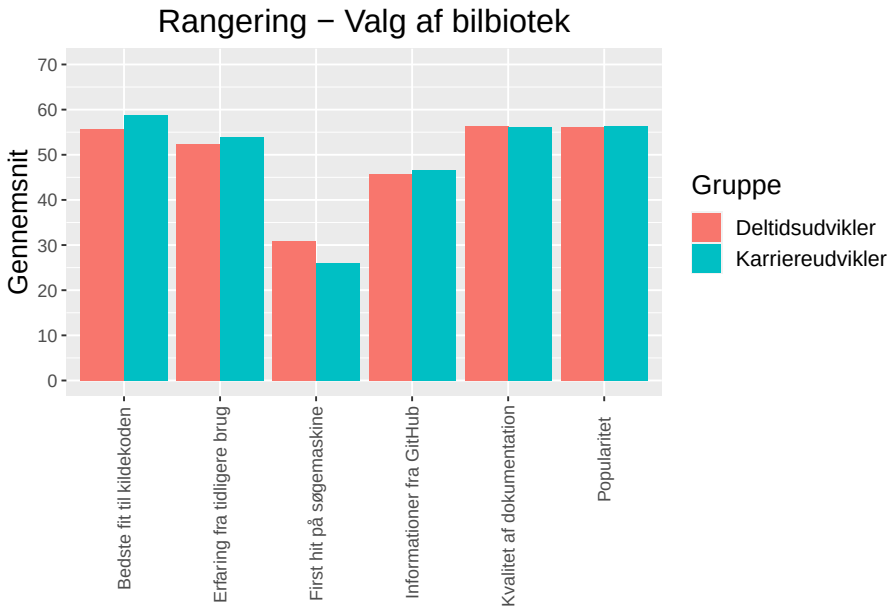
På figur 3.8 ses det at både Deltids udviklere og karriereudviklere rangerer "Worked-examples/kode snippets" og "Nem setup" højst, hvor "Dokumentation i kildekoden" bliver rangeret lavest af begge grupper.

Udregningen af denne rangering, samt de resterende rangeringsopgaver i spørgeskemaet er udført på samme måde som forklaret i appendiks B.1, ligning B.1.



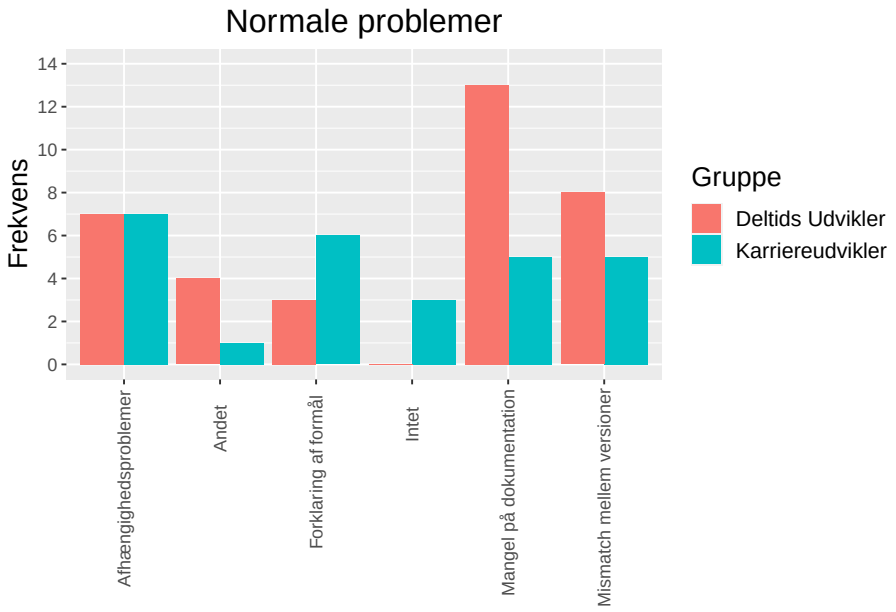
Figur 3.9. Figuren viser hvordan de to grupper "Deltids Udvikler" og "Karriereudvikler" har rangeret ift. indhold i dokumentationen. Henad x-aksen er de valgte emner og opad y-aksen er gennemsnittet. For at se de eksakte værdier se tabel B.8 og B.9.

På figur 3.9 ses det, at begge grupper vægter “Beskrivelse af formålet” og “En oversigt over brugen af biblioteket” højst, hvor grafiske elementer bliver rangeret lavest af begge grupper.



Figur 3.10. Figuren viser rangeringen for de to grupper "Deltids Udvikler" og "Karriereudvikler", som omhandler faktorer der influere i deres valg af bibliotek. Henad x-aksen er de valgte emner og opad y-aksen er det procentmæssige frekvens. For at se de eksakte værdier se tabel B.10 og B.11

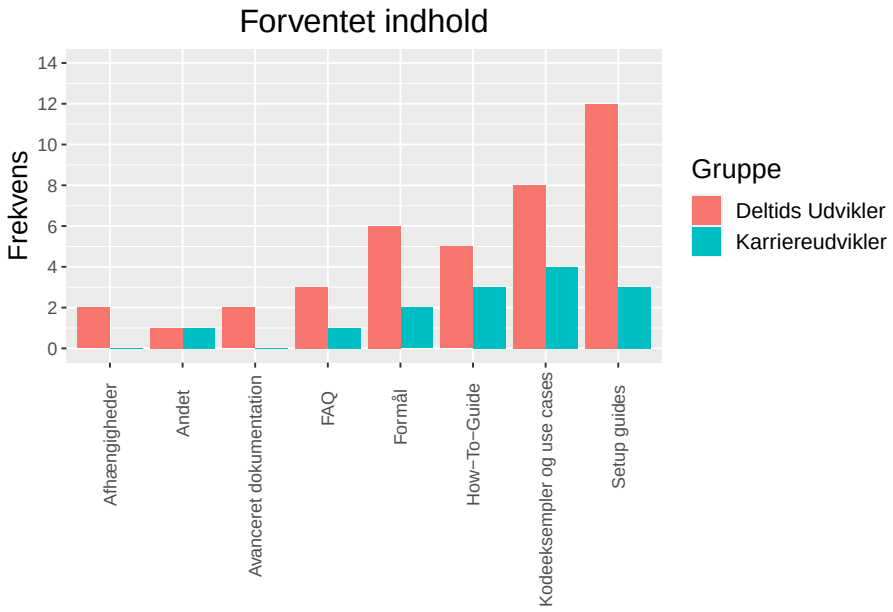
Det ses på figur 3.10, at de højeste rangeret faktorer der influrerer i valget af bibliotek er “Bedste fit til kildekoden”, “Kvalitet af dokumentation” og “Popularitet” for begge grupper. Den lavest rangeret faktor for begge grupper er “First hit på søgemaskine”.



Figur 3.11. Figuren viser fordelingen af de to grupper "Deltids Udvikler" og "Karriereudvikler" ift. hvilke problemer de er almindelige, når de bruger trejdepartsbiblioteker. Henad x-aksen er de valgte emner. Opad y-aksen er det frekvensen repræsenteret.

Til spørgsmålet om hvilke almindelige problemstillinger der opleves i brugen af trejdepartsbiblioteker, ses det på figur 3.11, at 13 af Deltids Udviklerne oplever mangel på dokumentationen. Dette gør det til den mest gængse problemstilling for denne gruppe. Otte af Deltids Udviklerene, har oplevet problemer med mismatch imellem versionerne, hvilket gøre det til den næsthøjeste problemstilling.

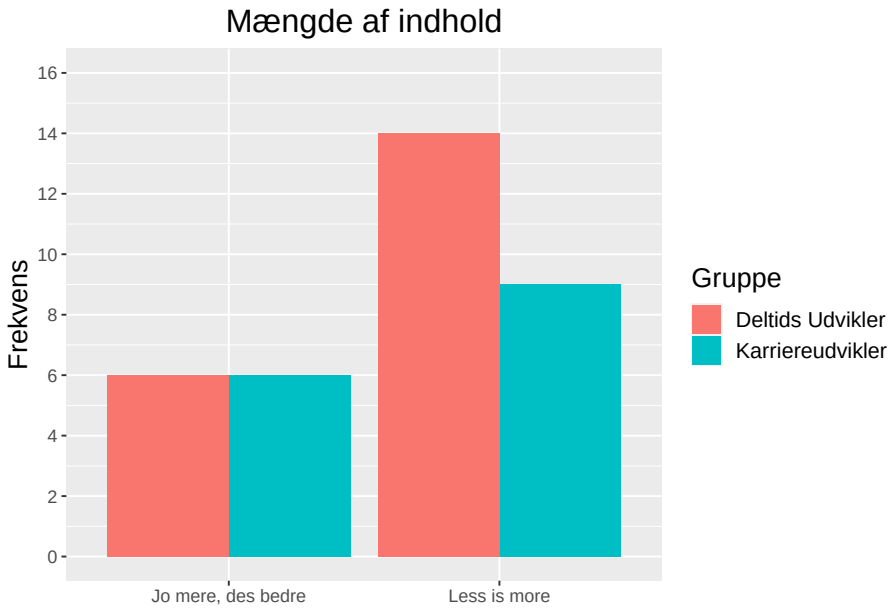
Karriereudviklerne har derimod primært valgt afhængigshedsproblemer (oversat fra *dependency issues*), som det mest gængse problem, da syv af respondenterne i denne gruppe har valgt dette. Sekundært har samme gruppe valgt, at der opleves problemer med forklaring af formålet med biblioteket, hvilket er seks af respondenterne, der har valgt.



Figur 3.12. Figuren viser fordelingen af de to grupper "Deltids Udvikler" og "Karriereudvikler" ift. hvad grupperne forventer af indhold i dokumentation. Henad x-aksen er de valgte emner. Opad y-aksen er det frekvens repræsenteret.

På figur 3.12 ses det, at fire respondenter i gruppen "Karriereudvikler" forventer at finde kodeeksempler og use cases i dokumentationen, hvilket gør det til det primære valg for denne gruppe. Sekundært er både "how to" guides og "setup" guides valgt med hver tre respondenter.

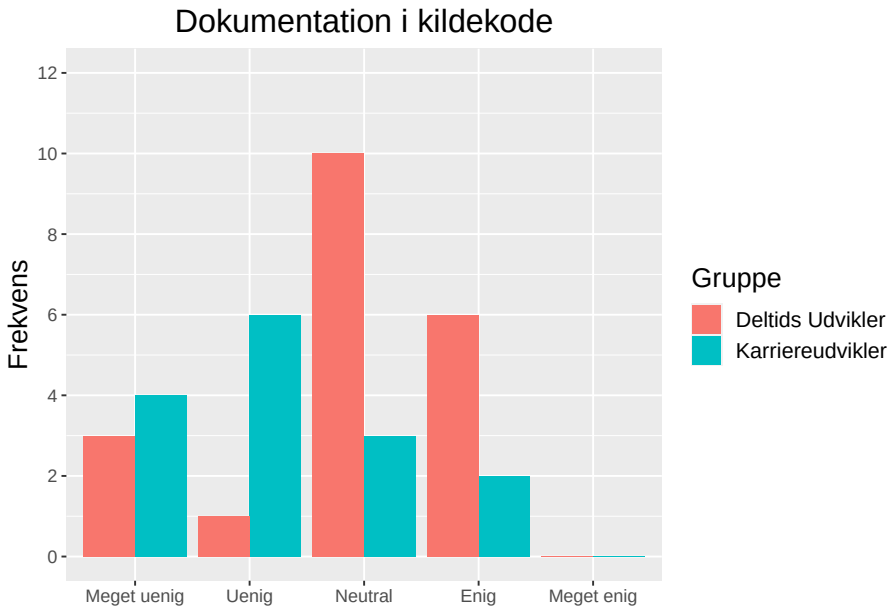
12 respondenter i gruppen "Deltids Udvikler" forventer setup guides i dokumentationen, hvilket gør til det primære valg og sekundært forventer otte respondenter kodeeksempler og use cases i dokumentationen.



Figur 3.13. Figuren viser fordelingen af de to grupper "Deltids Udvikler" og "Karriereudvikler" ift. hvordan man skal prioritere mængden af indhold i dokumentationen. Henad x-aksen er de valgte emner og opad y-aksen er det frekvensen repræsenteret.

Det ses på figur 3.13, at begge grupper prioritere *Less is more*-princippet over, at der skal være meget indhold i dokumentationen. Ni karriereudviklere har prioriteret dette princip højere end "Jo mere, des bedre". 14 Respondenter i gruppen "Deltids Udviklere" har også valgt *Less is more*-princippet over, at der skal være meget indhold i dokumentationen.

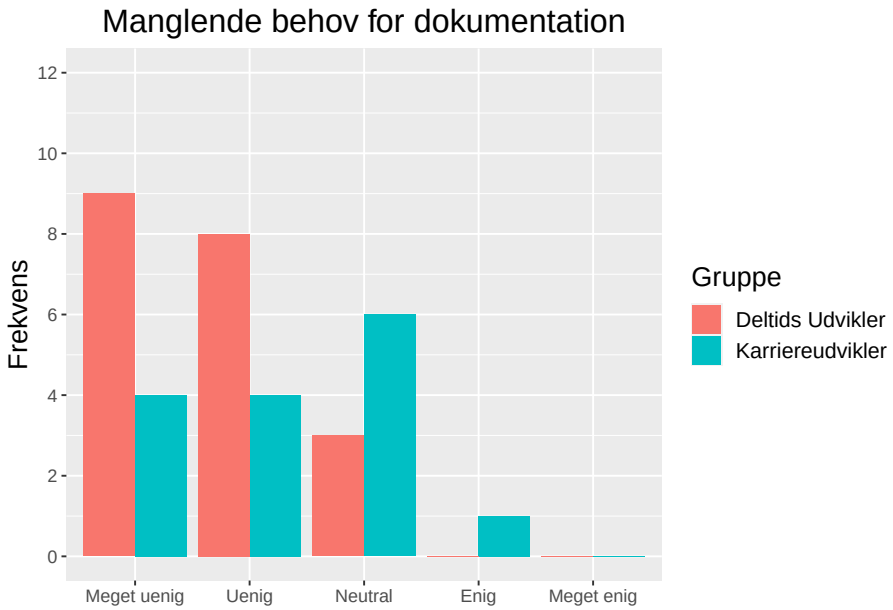
På de efterfølgende figurer har respondenterne haft til opgaven at afgive, hvor enige respondenten er i en række udsagn. Skalaen som respondenten er blevet præsenteret for, har været gående fra "Meget uenig" til "Meget enig". Se forklaring i appendiks B.1.



Figur 3.14. Figuren viser fordelingen af de to grupper "Deltids Udvikler" og "Karriereudvikler" ift. hvor enig man er i udsagnet om dokumentation i kildekoden. Henad x-aksen er de valgte emner og opad y-aksen er det frekvensen repræsenteret.

Ift. udsagnet "The main part of the documentation should be comments in the source code" har gruppen "Karriereudvikler" primært stillet sig uenig i dette udsagnn.

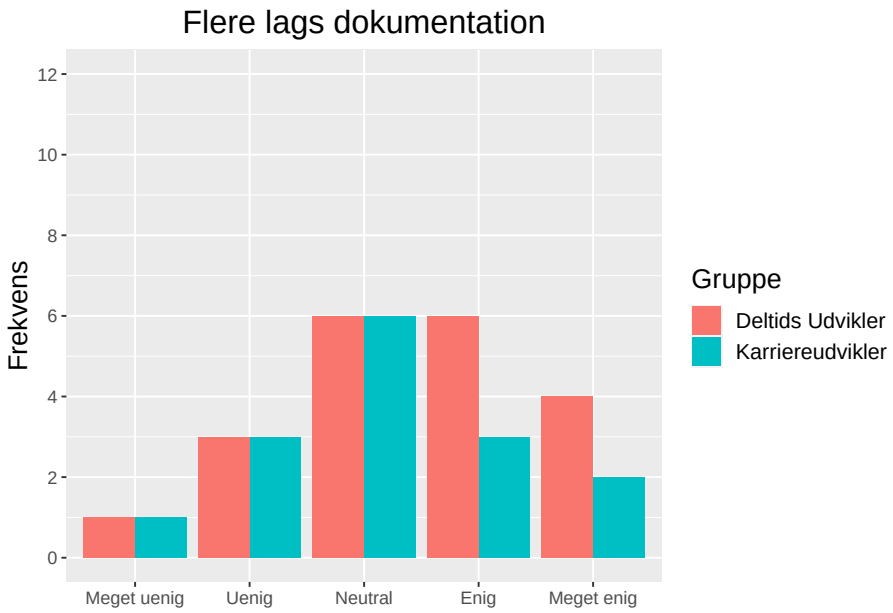
Gruppen "Deltids udvikler" har med 10 respondenter primært stillet sig neutralt til udsagnet. Sekundært har gruppen "Karriereudvikler" været meget uenige med udsagnet, hvor Deltids udvikler sekundært har været enige.



Figur 3.15. Figuren viser fordelingen af de to grupper "Deltids Udvikler" og "Karriereudvikler" ift. hvor hvor enig man er i udsagnet om manglende behov for dokumentation. Henad x-aksen er de valgte emner og opad y-aksen er frekvensen repræsenteret.

Som det ses på figur 3.15, har halvdelen af karriereudviklerne enten været meget uenige eller uenige i følgende udsagn: *No need for documentation because good code documents itself.* Seks respondenter i samme gruppe har derimod stillet sig neutrale ift. dette udsagn.

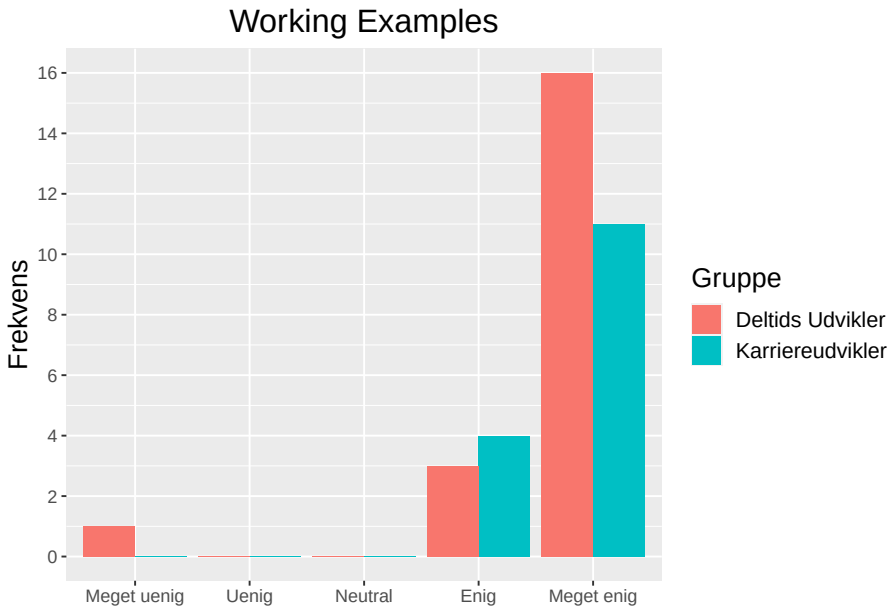
17 respondenter i gruppen "Deltids udviklere" har enten været meget uenige eller uenige i udsagnet og tre respondenter har forholdt sig neutralt.



Figur 3.16. Figuren viser fordelingen af de to grupper "Deltids Udvikler" og "Karriereudvikler" ift. flere lags dokumentation. Henad x-aksen er de valgte emner og opad y-aksen er det frekvensen repræsenteret.

Figur 3.16 ses det, at seks respondenter i gruppen "Karriereudvikler" har stillet sig neutralt til følgende udsagn: *A well documented library has minimum two layers of documentation, one for the beginner and one for advanced use.*

Der er ligeledes seks Deltids udviklere, som har forholdt sig neutralt til dette udsagn. Halvdelen i gruppen Deltids udviklere har enten været enige eller meget enige i udsagnet, hvor dette gælder for ca. 1/3 af gruppen "Karriereudvikler".



Figur 3.17. Figuren viser fordelingen af de to grupper "Deltids Udvikler" og "Karriereudvikler" ift. hvor enige de har været i udsagnet om working examples. Henad x-aksen er de valgte emner og opad y-aksen er det procentmæssige frekvens.

Som det ses på figur 3.17, er det gældende for begge grupper, at over 94 % af respondenterne har enten været enige eller meget enige i følgende udsagn: *A well documented library has working examples.*

3.2.3 Diskussion

Følgende afsnit har til formål at diskutere de begrænsninger som spørgeskemaet måtte have, samt diskutere de fundne resultater og årsagen til disse.

Det har primært været respondenter, der er bosiddende i Danmark, som har besvaret spørgeskemaet. Dette er på trods af, at spørgeskemaet kun har været sendt ud til én Facebook gruppe, som var henvendt til dansk talende personer. På de resterende facebook grupper foregik kommunikationen på engelsk og derved må det antages, at de sider henvendte sig til en bredere målgruppe. At der primært har været respondenter bosiddende i Danmark, kan have påvirket, at respondenterne ikke har været repræsentativ for hele målgruppen. Denne faktor anses dog kun som en mindre problemstilling, da ud fra resultaterne fra interviewsene (se afsnit 3.1) blev klarlagt, at udviklere generelt arbejder i et internationalt miljø og respondenterne derved har kendskab til normalen i engelsk talende miljøer. Dog kan man argumentere for, at det havde været fordelagtig, at have sørget for en større diversitet ift. hvor respondenterne bor, for at sikre en at resultaterne er retvisende for populationen.

Der var i alt 41 respondenter der besvarede spørgeskemaet, hvor otte af respondenterne kun delvist gennemførte spørgeskemaet. Disse 41 respondenter blev opdelt i to grupper, hvor der var samlet 23 respondenter i gruppen Deltids udvikler og 19 respondenter i gruppen "Karriereudvikler". Det lave antal respondenter i hver gruppe har gjort, at inferential statistik ikke er blevet vurderet relevant. Dette er grundet, at spørgeskemaer generelt

er opstillet op som et *between subject design*. Derfor vil det kræve et væsentlig højere antal respondenter for, at kunne undersøge om der er signifikant forskel i holdninger på de to grupper. Dog anses det stadig muligt at undersøge de to grupper for tendenser og via en sammenligning med resultaterne fra interviewsene, danne et grundlag for de emner der kræver opmærksomhed, når man designer for erfarne og uerfarne brugere i samme design.

Ud af de tre rangeringsopgaver som respondenterne blev stillet i spørgeskemaet, ses der ikke forskel i rangeringsorden for de to grupper. Man kan dog diskutere om de to grupper har samme forventninger til indhold, struktur og design.

På figur 3.7 ses det f.eks. at de to grupper af respondenter tendenser til ikke have samme prioritet angående dokumentationsform. Her har karriereudviklere primært valgt online manualler, hvor Deltids udvikler derimod primært fortrækker billeder med forklarende tekst. Her kan det argumenteres for, at dette har været misvisende sammenligning, da de online manualer kan indeholde billeder med forklarende tekst. Derfor kan respondenten have enten misforstået hensigten med emnet eller troet at de havde svaret for begge, når de har valgt det ene. Det ses dog, at karriereudviklere har valgt billeder med forklarende tekst som deres tertiære valg, da et design med kun tekst stadig er af højere prioritet af målgruppen.

Analysen viste, at den næstmest benyttede fejlsøgnings metode for begge grupper er, at kigge i dokumentationen til biblioteket. Analysen viste ydermere, at det mest normale problem for Deltids udvikler er mangel på signede dokumentation. Dette kan vidne om, at dokumentation generelt er skrevet mere til karriererudviklere end folk der ikke til dagligt arbejder med programmering.

Begge grupper af respondenter, har begge primært valgt kodeeksempler og setup-guides, som det forventede indhold i et bibliotek. Dog har karriereudviklerne valgt *how to* guides ligeså mange gange som setup guides. På baggrund af dette kan man argumentere for, at det er to essentielle emner for begge grupper i dokumentation af biblioteker.

Gillham [2008][s. 8] nævner i sin bog, at en af ulemperne ved at benytte sig af et spørgeskema er, at misforståelser af spørgsmål ikke kan blive rettet, som de f.eks. vil kunne blive under et interview. Derfor kan det diskuteres om spørgeskemaet har haft lav intern validitet, da man ikke kan være sikker på, at man måler på det der er hensigten. Et eksempel på dette, kan f.eks. være udsagnet om flere lags dokumentation (se figur 3.16), da 30 % eller derover har været neutrale for begge grupper. Dette kan være pga. at konteksten som spørgsmålet er udarbejdet under (se afsnit B.1) ikke er synligt for respondenterne. Den mulige misforståelse kan være sket på trods af, at spørgeskemaet og derved spørgsmålene, har været igennem to iterationer med tre udviklere, hvor formålet var at sætte ordlyden i spørgsmålene korrekt. Spørgsmålet er ydermere blevet udformet på baggrund af interviewsene, der er udført tidligere, hvor der er brugt samme vendinger i spørgsmålet, som der blev gjort under interviewsene.

3.2.4 Opsummering af væsentlige resultater

Dette afsnit vil opsummere de væsentlige resultater, der vurderes relevante for den videreudvikling.

Resultaterne fra spørgeskemaet viste, at de fleste Deltids udviklere oplever mangel på dokumentation til deres biblioteker, hvor "Karrerudviklere" primært oplever afhængighedsproblemer (oversat fra *dependency issue*).

Deltids udviklere fortrækker ydermere at finde dokumentationen på en dedikeret hjemmeside, hvor majoriteten af "Karrerudviklere" ønsker at finde den på GitHub eller lignede platforme.

Resultaterne fra spørgeskemaet viste en generel tendens til at *how to* guides, *working examples* og setup guides er prioritet højt, når det gælder indhold i dokumentationen af begge grupper af respondenter. Ydermere er forklaring af formål, samt brug af funktioner og variabler af høj prioritet for respondenterne i begge grupper, da dette er blevet nævnt andenflest gange.

Som diskuteret i afsnit 3.2.3 er de to grupper uenige i deres fortrukket dokumentationsform. Hvor Deltids udvikler har nævnt billeder med forklarende tekst flest gange, så har karriereudviklerne nævnt online manualler flest gange.

Selv på det sekundære valg, er de to grupper ikke ens, da Deltids Udvikler har valgt video tutorials og karriereudviklerne har valgt dokumentation i ren tekst form her.

3.3 Konklusion på brugerundersøgelsen

Brugerundersøgelserne havde til formål at undersøge udviklernes behov, udfordringer og ønsker til dokumentation af trejdeparts biblioteker.

Den gennemgået teori i kapitel 2, interviews og spørgeskemaer har givet en indsigt i de forventninger og behov som udviklere har til dokumentationen for trejdeparts biblioteker. Ud fra brugerundersøgelserne kunne det udledes, at forventningerne til indhold overordnet set er det samme for erfarne og uerfarne udviklere. Dog vurderes det ud fra teorien og spørgeskemaet, at strukturen og designet af indholdet forventes at være forskellige for de to grupper.

Ydermere kunne det udledes, at de gængse udfordringer med brugen af tredjeparts biblioteker er mangel på dokumentation, forskelle i opbygning imellem versioner og afhængighedsproblemer. Ift. udviklingen af dokumentationen finder de interviewede dokumentationen svært at skrive til brugeren, så den er let forståelig for brugeren.

Følgende er en liste af de højst prioriteret emner, som er blevet udledt på baggrund af brugerundersøgelserne:

1. Setup guides
2. Working examples/kodeeksempler
3. How-To-Guides
4. Use cases
5. Komplette eksempler

Brugerundersøgelser, samt den gennemgået teori har ydermere resulteret i tre fokuspunkter. Det første fokuspunkt er, at dokumentationen skal kunne inkludere både erfarne og

uerfarnes brugers arbejdsmønstre. Her skal der være særlig fokus på, hvordan karriereudviklere og deltidsudviklere ønsker at modtage instrukser i brugen af dokumentationen. Et andet fokuspunkt i designet af dokumentation for tredjeparts biblioteker er forklaring af formål. Dette gælder både formålet med biblioteket, men også handlinger der skal udføres for at benytte biblioteket. Det tredje fokuspunkt er det minimalistiske design, hvor *less is more*. Med dette menes der, at der kun skal vises informationer som er essentielle for siden.

Brugerundersøgelsen besvarer dermed de første tre forskninsspørgsmål, som lyder således: "Forskningsspørgsmål 1: Hvordan anvender udviklere med forskellige programmeringsevner dokumentationen til tredjeparts biblioteker?", "Forskningsspørgsmål 2: Hvilke udfordringer opleves der i udviklingen af dokumentationen og i brugen af den?" og " Forskningsspørgsmål 3: Hvilke ønsker har udviklere med forskellige programmeringsevner til indhold i dokumentationen for trejdeparts biblioteker?".

Ud fra resultaterne fra brugerundersøgelsen vælges det derfor at udvikle en prototype, der har til formål at konkretisere designet for trejdeparts biblioteker, samt at besvare forskningsspørgsmål 4, som er beskrevet i afsnit 2.5.

Det vælges at afgrænse projektet til at fokusere udelukkende på forsiden, setup guides og *working examples*.

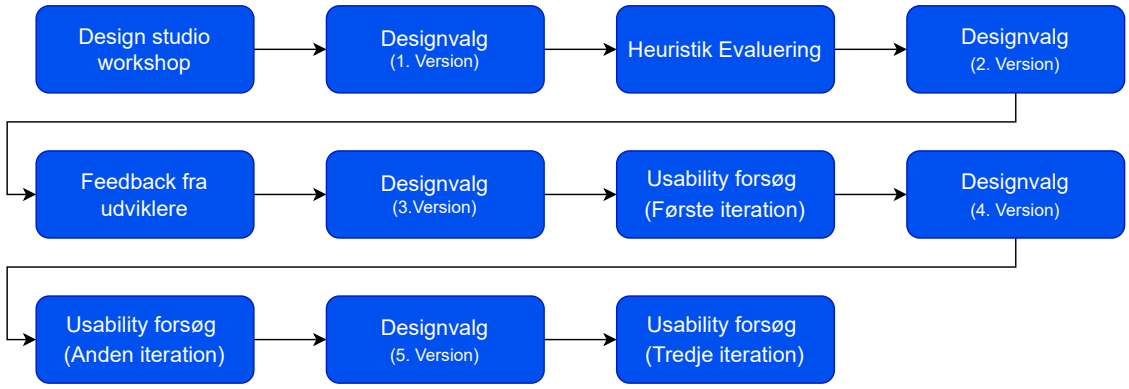
Målgruppen defineres til både at indeholde personer på hobbyniveau, der har et basal kendskab til programmering og professionelle udviklere.

Udformning af prototype 4

Følgende kapitel har til formål at beskrive designprocessen for udviklingen af dokumentation for trejdeparts biblioteker.

I designprocessen vil der blive taget udgangspunkt i Seluxits hjælper-bibliotek WappstoIoT, som har til formål sende og modtage data fra ens enhed, samt automatisere handlinger for ens enheder. Dette sker via IoT platformen Wappsto. Biblioteket har ligeledes til formål at give udviklere mulighed for at udvikle webbaseret applikationer i Wappsto.

En gennemgang af de fornødne steps for opsætning af disse biblioteker kan findes i appendiks E.



Figur 4.1. Figuren viser de iterationer, der er fortaget i udviklingen af prototypen.

På figur 4.1 ses det overordnet flow for udvikling af prototypen. I dette kapitel vil "Design Studio Workshop", "Heuristisk Evaluering" og "Feedback fra udviklere", samt de tilhørende prototype versioner blive gennemgået. I kapitel 5 vil "Usability forsøgene" blive gennemgået, samt deres tilhørende prototype versioner.

4.1 Design studio workshop

Dette afsnit har til formål at gennemgå den workshop, der ledte op til den første version af prototypen, samt argumentere for de valg, der er truffet i designet.

Der vælges at udføre en design studio workshop, da denne type workshop inkorporerer brainstorming, kritik og prioritering i én session. En af fordelene ved denne type workshops er ifølge Kate Kaplan [2017], at man kan samle personer med forskellige baggrunde og perspektiver til at løse en fælles problemstilling. Ydermere kombinere workshoppen den

divergente og konvergente tænkning [Kate Kaplan, 2017].

Målet med denne workshop er derfor, at udarbejde struktur og på skitseniveau have illustreret fokussiderne: Forsiden, setup guide og *Working example*. Skitserne for fokussiderne skal dernæst bruges som inspiration til første version af prototypen.

Det benyttede PowerPoint, samt videooptagelser og rådata kan findes i bilag B.

Fremgangsmåde

Denne workshop vil blive afholdt med inspiration fra Kate Kaplan [2017] forslag til struktur af design studio workshops.

Figur 4.2. Figuren viser de fem faser som workshoppen er inddelt i.

Som det ses på figur 4.2, er workshoppen opdelt i fem faser; Introduktion, Skitsere, Præsenter og kritisere, Sammenflet og Prioriterer.
Den resterende del af afsnittet vil gennemgå hver fase og dens indhold og fremgangsmåde kronologisk.

Fase 1 - Introduktion
I den indledende fase af workshoppen får deltagerne en introduktion til workshoppen, samt et indblik i de mest væsentlige resultater fra brugerundersøgelsen og den gennemgået teori. Dette vælges, da deltagerne ikke har været en del af denne proces og det ønskes at skabe en fælles forståelse for de hidtidige resultater. Til sidst i introduktion vil deltagerne ligeledes blive informeret omkring hvilken problemstillinger, de skal arbejde med. I denne introduktion vil de deltagende også blive instrueret i, at ingen ideer er forkerte og der vil blive forsøgt at skabe et frit miljø, hvor alle får mulighed for at bidrage.

Fase 2 - Skitsere
Formålet med denne fase er først, at generere mange ideer på kort tid, hvorefter disse ideer skal konkretiseret og skitseret i en mere detaljeret form. Til sidst i denne fase, skal deltagerne ligeledes overveje for ruten fra forsiden og ud til de to andre fokussider.
Under hele fase 2 arbejder de deltagende individuelt og vil arbejde med *Working example*, setup guides og forsiden.

Metoden der benyttes i denne fase kaldes "Crazy 8's" og er opdelt i tre runder. I den første runde vil deltagerne få udleveret et stykke A4 papir, hvor der er otte lige store felter. Dernæst vil deltagerne få til opgave, at komme på otte ideer på otte minutter. Her vil de blive instrueret i, at det er strukturen på siderne der har interesse og at der derfor ikke skal udarbejdes et detaljeret.

Når de otte minutter bliver sat i gang, vil deltagerne arbejde med *working examples* i de første tre minutter, dernæst vil de arbejde med forsiden i de efterfølgende tre minutter. I de sidste to minutter vil deltagerne skulle arbejde på setup guiden.

I anden runde får deltagerne otte minutter til at skitsere hver fokusside på et mere detaljeret

plan. Derfor skal de også benytte et helt A4 papir for hver fokusside. Inden påbegyndelse vil deltagerne blive instrueret i, at skulle bygge videre på første rundes ideer, som de kan kombinere som de ønsker.

I tredje runde har deltagerne fem minutter til at skitsere vejen fra forsiden til de to andre fokusside, hvor de skal gøre det så detaljeret, som det er muligt indenfor tidsrammen.

Facilitatoren vil under hele denne fase have til opgave at styre tiden og vejlede deltagerne hvornår og hvad de skal gøre under hver runde.

Fase 3 - Præsenter og kritisere
Formålet med denne fase er, at dele og diskutere ideer igennem en konstruktiv kritik.

I denne fase får deltagerne til opgave, at præsentere deres skitsering, som de har lavet i runde to og tre i fase to. Dernæst vil deltagerne blive instrueret i at give feedback indenfor specifikke emner.

For at holde feedbacken struktureret og konstruktiv vil er framework, som er udviklet af Edward de Bono Group [2017] kaldet *Thinking hats* blive benyttet. Dette framework bruger seks metaforiske hatte. Hver hat repræsenterer forskellige måder at tænke på, hvilket har til formål at få deltagere til at tænke og diskutere ideer effektivt sammen indenfor faste rammer. I denne fase skal de stærkteste ideer ved skitserne identificeres og skrives ned. Der sættes et tidsinterval på et minut for hver hat og facilitatoren vil guide deltagerne videre til næste hat, når det tiden er gået.

Fase 4 - Sammenflet
Formålet med denne fase er, at deltagerne skal tage de stærkeste elementer fra tidligere faser og kombiner dem til nye samlet idéer. Dette gøres ved fælles gruppearbejde, hvor den maksimale tid er sat til 10 minutter. Gruppen skal forsætte brainstormen og forsøge at drive visionen fremad sammen. Når der er skabt forskellige skitser med de stærkeste elementer forsættes der til næste fase.

Fase 5 - Prioriterer
Formålet med denne fase er, at definerer og prioriterer fælles temaer og elementer skitserne imellem. Dette har til formål at skabe konsensus i gruppen om fælles temaer og de elementer, der har højeste prioritet. Måden hvorpå dette undersøges er via $100 test, hvor deltagerne får et fiktivt pengebeløb, de må placere ved de forskellige elementer og skitser. Dette gøres uden gruppen må snakke og beløbene skrives ned på et papir for at undgå at deltagerne påvirker hinanden.

Efter den femte fase af workshoppen er målet, at der er udarbejdet en struktur og design for forsiden, setup guide og *working examples*. Dette skal herefter overføres til en prototype, som så kan videreudvikles på.

Pilotforsøg

Grundet de mange forskellige facetter og faser i workshoppen anbefaler Kate Kaplan [2017], at facilitatoren skal være komfortabel med at lede workshoppen og det kræver ifølge Kate Kaplan [2017] erfaring. Derfor vælges det også at udføre et pilotforsøg med tre Produkt- og Designpsykologi studerende fra Aalborg Universitet. Disse personer er

valgt, trods begrænset erfaring med programmering, da formålet med pilotforsøget er at give facilitatoren erfaring i afholdelse af workshoppen, samt modtage feedback på gennemførelsen af workshoppen.

Det benyttede PowerPoint, samtykkeerklæringer, lydoptagelse og skitser mm. kan findes i bilag B i mappen "Pilotworkshop".

Da deltagerne ikke er velkendte i programmering og brug af tredjeparts biblioteker, er der udarbejdet en ekstra introduktion til pilotforsøget, hvor disse aspekter vil blive forklaret. Pilotforsøget vil ydermere foregå på dansk, hvorimod den endelige workshop vil foregå på engelsk. Dette er valgt i forsøget på at undgå at begrænse deltagernes feedback pga. sprogbarriere. Skitser og andre resultater vedrørende design og struktur fra dette pilotforsøg har ikke interesse for det videre designforløb, da dette pilotforsøg udelukkende skal ses som rutenering af facilitatoren, samt at udlede fejl og mangler i forsøgsdesignet.

Pilotforsøget blev gennemført i et grupperum på Aalborg Universitet, hvor der var TV, tavler og papir til rådighed for de deltagende. Selve workshoppen blev udført som gennemgået i afsnit 4.1. Efter hver fase af workshoppen, blev deltagerne bedt om at give feedback på metodevalget og komme med forslag til mulige alternativer til det valgte.

Resultater fra pilotforsøg
De væsentlige resultater fra pilotforsøget er følgende:

1. Facilitatoren skal være opmærksom på ikke at benytte fagtermer, som deltagerne muligvis ikke har forståelse for som f.eks. *user stories* og *personas*.
2. I de givne eksempler af biblioteker, må der gerne være større forskel på disse, så man viser forskellige typer brug af biblioteket.
3. Sørg for at have klargjort foldet papir inden metoden "Crazy 8's" skal bruges, for at minimere den brugte tid.
4. Opfordre deltagerne til at bruge al tid, der givet på at optimere deres ideer, så de ikke sidder stille. Dette gælder især fase 1.
5. Få formidlet meget tydeligt til deltagerne, at det ikke forventes at workshoppen ender ud i en komplet prototype, men skal mere ses som at skabe grundlag for en prototype.

Deltagerne til pilotforsøget havde inden start fået en tidsestimering på en time, hvilke viste sig at være for lidt. Da to af deltagerne havde anden aftale blev fase fem, blot forklaret og diskuteret. Tidsestimeringen til den endelige workshop vil være på halvanden time.

Som facilitator er det blev erfaret, at for at udføre denne workshop indenfor en acceptabel tidsramme, er det nødvendigt med en stram tidsstyring. Ydermere er det blevet erfaret at informationer såsom resultater fra brugerundersøgelser skal være tilgængelige under hele workshoppen, så de deltagende hele tiden kan verificere deres skitser med disse.

4.1.1 Resultater

Den endelige workshop blev udført i et mødelokale ved Seluxit, hvor deltagerne deltog fysisk.

I mødelokalet var der adgang til whiteboards, projektor og et flipoverstativ. Workshoppen tog ca. 1 time og 50 minutter inklusiv en fem minutters pause. Optagelser, skitser,

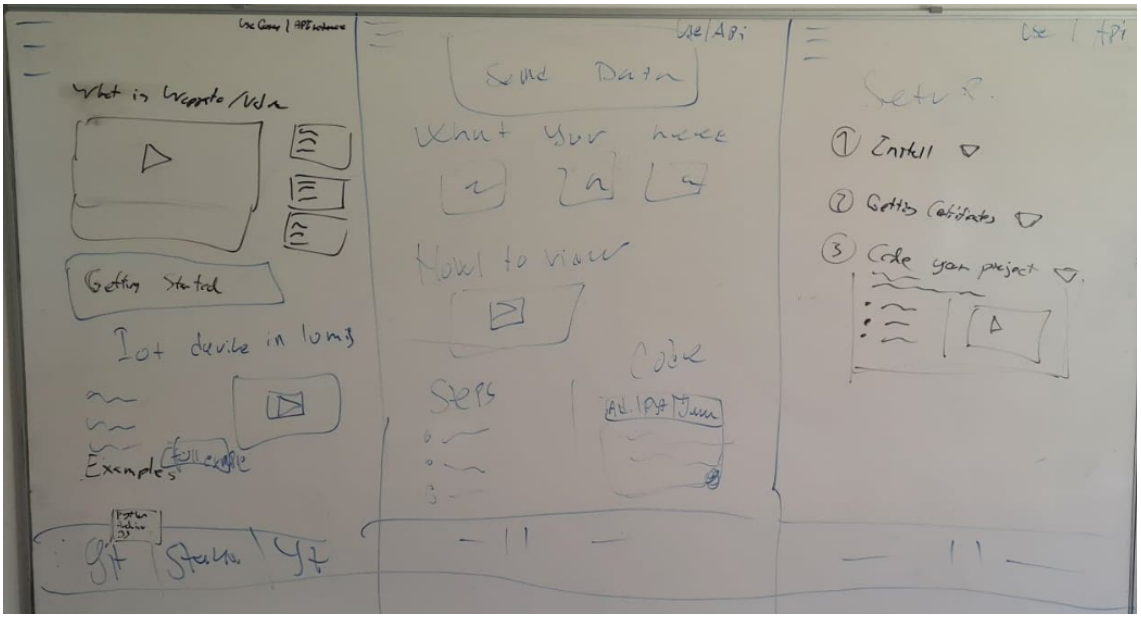
samtykkeerklæring og hjælpemidler, som blev brugt til at udføre workshoppen, kan findes i bilag B i mappen "Workshop".

Dette afsnit vil kun præsentere resultaterne fra fase 4 og 5. Resultaterne fra de tidligere faser kan findes i bilag B i mappen "Workshop". Resultaterne er blevet udarbejdet via en manuelt udført videoanalyse, hvor sessionen er blevet gennemgået og resultaterne er blevet noteret ned.

Navn	Jobtitel	Erfaring med programmering	Primære programmeringssprog
Simon	Salgschef	Deltidsudvikler	Blockly, Python
Tvsetomir	Backend programmør	Karriereudvikler	Python, JavaScript
Mads	Product Manager	Deltidsudvikler	R, JavaScript, Python

Tabel 4.1. Tabellen viser de tre deltagendes programmeringserfaring, jobtitel og primære programmeringssprog.

Som det ses på tabel 4.1, har deltagerne forskellige erfaringsningsgrundlag med programmering og arbejder i forskellige afdelinger i Seluxit. Dette er et bevidst valg, da det netop er en af styrkerne ved denne type workshops, da man ved at inddrage personer med forskellige baggrunde, kan inddrage forskellige holdninger og perspektiver i diskussionen og derved udviklingen af prototypen [Kate Kaplan, 2017].



Figur 4.3. Figuren viser respondentenes tegning i fase 3 - Sammenflet. Her er forsiden placeret længst til venstre, *working example* i midten og setup guide yderst til højre. Siderne er adskilt med en vertikal streg.

Som det ses på figur 4.3, resulterede sammenfletningen fra fase 4 ud med tre skitser for strukturen og det overordnet indhold for siderne: forsiden, *working examples* og setup.

Fælles for alle sider er header og footer. Som det ses på figur 4.3 er headeren designet således, at man kan tilgå Wappsto platformen direkte, og use cases kan ligeledes tilgås via headeren. Ydermere er der designet i venstre side for en menu, der skal kunne åbne hvis der trykkes. Her vil man som bruger kunne få et overblik over alt indhold i dokumentationen.

Footeren er designet således, at man kan tilgå GitHub, Stack Overflow og YouTube direkte via links.

På forsiden, som ses yderst til venstre på figur 4.3, er der skitseret for en overskrift med titlen "What is Wappsto?". Overskriften bliver efterfulgt af en video, der har til formål at forklare hvad Wappsto er. I tilknytning til videoen, er der skitseret for en forklarende tekst i højre side af videoen. Dernæst designede deltagerne for en "Getting started" knap, der har til formål at navigere brugeren over til setup siden. Denne knap bliver efterfulgt af en overskrift lydende "IoT device in 10 min.". Hvorefter der er designet for et tekstfelt og en video, der har til formål at eksemplificere, hvordan man kan få en enhed forbundet til internettet. Man får ligeledes muligheden for at se det fulde eksempel, da der er designet for en knappe umiddelbart efter eksemplet.

Som det ses på figur 4.3, er der øverst på eksempel siden designet for en overskrift. I dette eksempel tages der udgangspunkt i overskriften "Send Data". Dernæst får brugeren informationer om, hvad man skal bruge for at kunne gennemføre dette eksempel. Dernæst får brugeren mulighed for at se en video, der omhandler hvordan man udarbejder eksemplet. Dette forklares efterfølgende yderligere, hvor koden er placeret til højre på siden og til venstre forklares hvert nødvendigt step. I koden, har brugeren mulighed for at vælge sit fortrukne programmeringssprog.

Setup siden er designet således, at der øverst på siden er designet for overskriften "Setup", hvorefter man kan se tre steps. Første step kaldes "Install", andet step kaldes "Getting certificates" og tredje step kaldes "Code your project". Alle tre steps kan foldes ud ved at trykke på dem. Dette er eksemplificeres ved step 3, hvor man kan se, at der er designet for en video med tilhørende forklarende tekst.

Tsvetomir		Mads		Simon	
Hvad	Kr.	Hvad	Kr.	Hvad	Kr.
Struktur af forsiden	30	Formål på forsiden	20	Forsiden	20
Setup guide tekst og struktur	20	Footer	20	Eksempler	60
Eksempel videoer	20	Setup	25	Setup	20
Eksempel tekst	10	Use Case	25		
Footer	5				
Side menu	15				

Tabel 4.2. Den tabel viser hvilke emner deltagerne har prioriteret og hvilken værdi de har givet hvert emne.

På tabel 4.2 ses det, at Tsvetomir har prioriteret strukturen af forsiden højest, hvor Mads har prioriteret hele setup siden og use case højst. Simon har vurderet eksempler tre gange højere end både forsiden og setup. Teksten ved Setup guiden og dennes struktur har Tsvetomir vurderet anden højest, sammen med videoerne til eksemplerne. Det sekundære valg for Mads var både formål på forsiden og footer. Simons anden prioritet var ligeledt fordelt på forsiden og opsætningsguiden.

4.1.2 Diskussion

Følgende afsnit har til formål at diskutere resultaterne fra workshoppen.

Forinden den endelig workshop, blev der udført en pilot workshop, der havde til formål at teste fremgangsmåden for workshoppen. Tre personer blev udvalgt, som værende deltagere. Disse personer blev udvalgt, da de alle har en forståelse for udvikling af prototyper og metoder indenfor workshops. Dog havde ingen af de deltagende indgående kendskab til hverken programmering eller tredjeparts biblioteker. Det kan derfor have været svært for de deltagende, at relatere sig til konteksten og de overvejelser, der skal medtages i udviklingen af et tredjeparts bibliotek. Dog var hensigten blot at få feedback på struktur og formidling og der vurderes derfor at den manglende kontekst ikke har været essentiel for workshoppen. Dog kunne man have gentaget pilotforsøget med personer, der har kendskab til programmering efter rettelserne fra det første pilotforsøg. Det vurderes dog at grundet tidsøkonomiske begrænsninger, samt et lavt antal tilgængelige forsøgspersoner, at dette ikke var muligt.

Både under pilotworkshoppen og den endelige workshop, blev der indledt med en opsummering af væsentlige resultater. Dog kunne man have sendt disse informationer ud forinden og derved minimeret den kognitive belastning for deltagerne under workshoppen. Dette vil ydermere have medført, at tiden for workshoppen vil blive kortere.

Under fase 3, fik de deltagende til opgave at kritisere hinandens skitser ud fra seks metaforiske hatte, som er udarbejdet af Edward de Bono Group [2017]. Det blev observeret at de deltagende generelt havde svært ved at holde sig indenfor de aktuelle hattes emner. Dette kan man diskutere om, det har været grundet at de deltagende i begge workshops kun fik en overordnet forklaring af hattens formål. Ydermere fik de blot en introduktion til hver aktuel hat, i den rækkefølge som de skulle bruge dem og derfor ikke en samlet gennemgang af dem. Grundet denne overordnet introduktion til denne metode, fik deltagerne ikke et dybere kendskab til formålet med værktøjet eller eksempler på at bruge dem. Man kan derfor argumentere for, at hvis deltagerne havde fået eksempler på brugen forinden ville de være i bedre stand til at kunne holde sig indefor det aktuelle emne.

Ud fra Kate Kaplan [2017] forklaring af *Crazy 8s*, skal man kun fokusere på en side i denne teknik. I den gennemførte workshop, blev der dog vekslet imellem tre forskellige sider, for at maksimere udbyttet af workshoppen. Dette kan man diskutere om det har haft en negativ indflydelse i opgaveløsningen, da der ikke er opnået det kvantitative udbytte, som hensigten er med metoden. Dog vurderes det ud fra resultaterne, at der er opstået ideer under denne fase, som deltagerne har kunne benytte senere i workshoppen. Man kan dog diskutere om man skulle have gentaget denne teknik tre gange for at få det ønskede antal sider med. Dette kunne dog have medført *fatigue effect* i et tidligt stadie og derfor have influeret til sidst i workshoppen.

Der var afsat 10 minutter til Fase 4, som dog blev overskredet med cirka 20 min. Det blev dog vurderet under workshoppen, at lade denne fase gå over tid, da deltagerne diskuterede konstruktivt konstant under denne fase. Derfor blev det vurderet, at ved at stoppe jf. den planlagte tid, vil man have forringet resultatet af denne fase.

Under den sidste del af workshoppen, hvor de skulle prioritere enkelte aspekter af deres skitse fra fase 4, havde to ud af de tre deltagere misforstået opgaven. Begge havde prioriteret hele eller store dele af skitserne. Det blev dog vurderet, at det ikke var muligt at gentage denne del, da deltagerne havde præsenteret deres prioritering for hinanden og

de derfor kan have påvirket hinanden i en eventuel anden runde.

4.1.3 Delkonklusion

Formålet med denne workshop var at udarbejde struktur og på skitseniveau have illustreret fokussiderne.

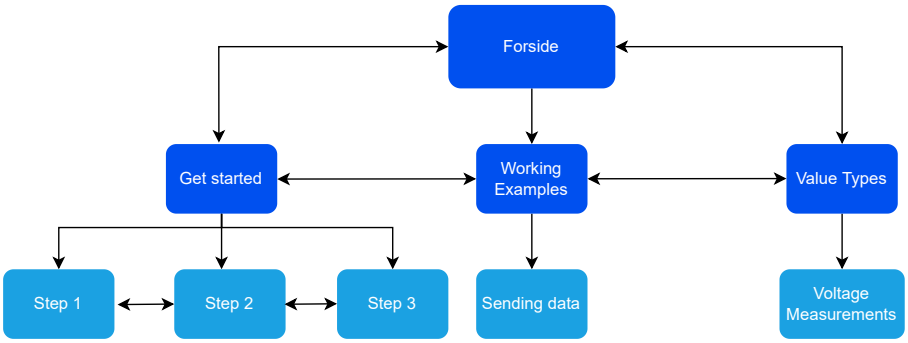
På trods af enkelte udfordringer med misforstået opgaver og tilrettelser af metoder ift. oprindelig brug, vurderes det at der er opnået et tilfredsstillende resultat. De udarbejdet skitser vil derfor danne grundlag for den første version af prototypen for dokumentation af tredjeparts biblioteker.

4.2 1. version

Dette afsnit har til formål at gennemgå første version af prototypen for dokumentation af tredjeparts biblioteket WappstoIoT.

Denne version af prototypen er udarbejdet i designværktøjet Adobe XD[1], da det ønskes at kunne fortage justeringer i designet hurtigt og nemt imellem iterationerne. Der bliver kun designet for programmeringssproget Python på dette stadie af prototypen.

Den gennemgået teori i kapitel 2 og brugerundersøgelserne gennemgået i kapitel 3 viste, at man skal påtænke, at brugeren skal have hands-on erfaring og aktiv involvering så tidligt i dokumentationen som muligt. Ydermere skal mængden af informationer holdes til det minimale. Derfor har disse emner været fokuspunkter i hele designprocessen for prototypen.



Figur 4.4. Figuren viser flowet for første version af prototypen.

Som det ses på figur 4.4 startes første version af prototypen på *Forside*, dernæst kan brugeren vælge at tilgå *Get Started*, *Working Example* eller *Value Types*. *Get Started* siden er opsætningsguiden, hvor man kan opsætte sit projekt til biblioteket. Hvis man er tilgået *Get Started* kan man tilgå alle tre steps direkte i opsætningen. På siden *Working Example* kan brugeren se seks forskellige eksempler, der er dog kun designet for ruten *Sending data* på nuværende stadie.

Value Types er et specifikt tilfælde for dette bibliotek og er derfor ikke i særlig interesse for projektet. Det er dog indledningsvis fundet nødvendigt, at brugere kan tilgå disse

[1] `https://www.adobe.com/dk/products/xd.html`

informationer, når de skal udføre forsøg. Der er derfor designet for typen *Voltage Measurements*. Formålet med *Value Types* er, at brugerne skal definere SI enheden, som deres enhed måler i. Dette er for at Wappsto kan behandle data input efter hensigten.

Ens for alle sider i 1. version af prototypen, er headeren og footeren, som blandt andet ses på figur 4.5. Fra headeren har brugeren mulighed for at tilgå *Get started*, *Examples* og *Use Cases* direkte. Dette er valgt grundet brugerundersøgelserne, der viste at disse tre emner var essentielle for brugere at kunne tilgå nemt (se blandt andet afsnit B.1).
Fra headeren kan man også tilgå *Value Types*, *API Documentation* og *Wappsto*. Det er valgt at placere disse med en mindre tekststørrelse og i øvre højre hjørne, da disse funktioner anses som at blive brugt færre gange end de øvrige kategorier i headeren. Ydermere er disse kategorier henvendt til den mere erfarne bruger. Der er ligeledes designet for en søgefunktion, da det via brugerundersøgelserne blev udledt at nem navigation var vigtigt for et veldokumenteret bibliotek (se figur 3.9). Det er hensigten at headeren vil følge med, når man scroller på siden. Denne funktionalitet er dog ikke muligt at implementere i Adobe XD.

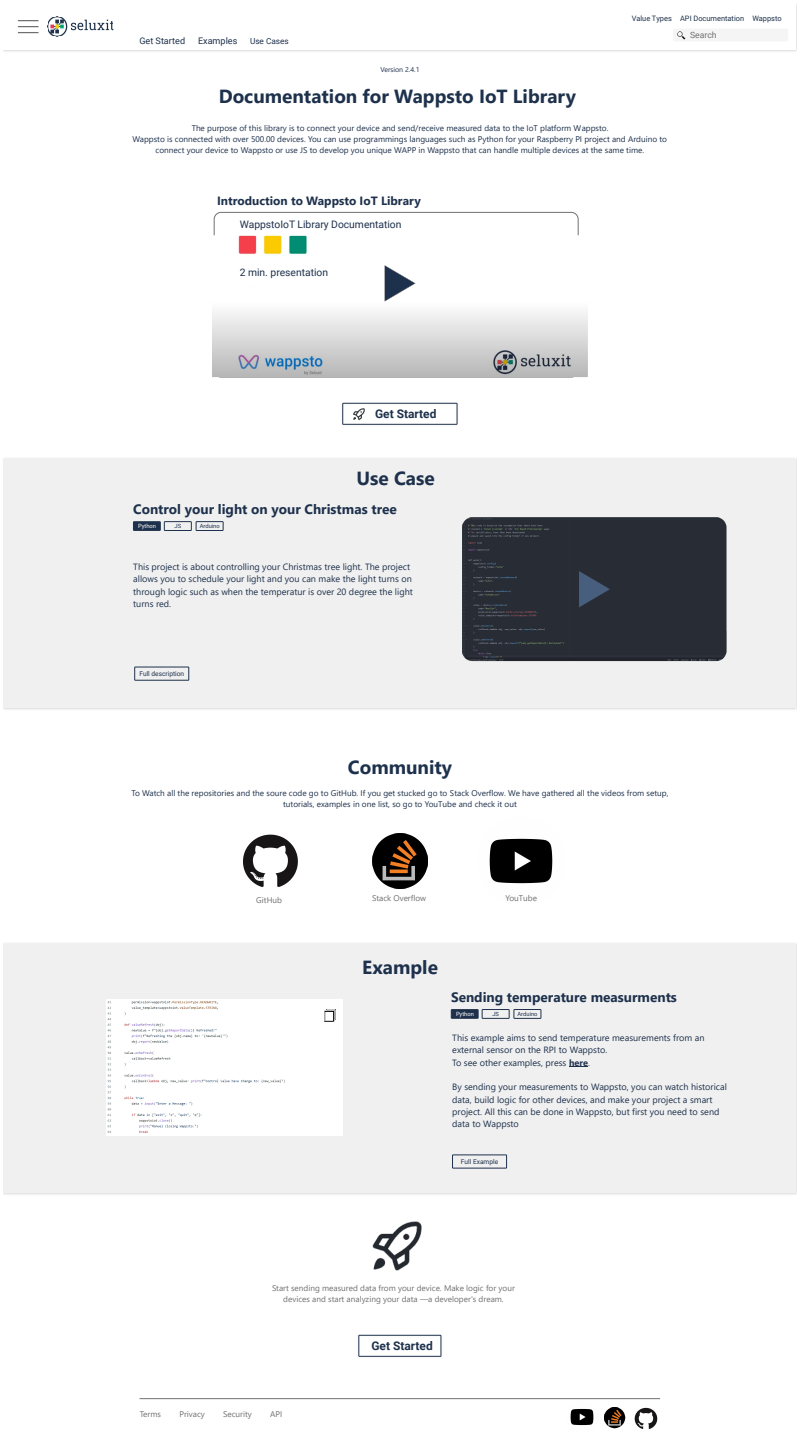
Footeren er designet således, at man kan tilgå de medier, der blev fundet relevante i brugerundersøgelserne (se figur B.21). Hvis man trykker på YouTube-ikonet, er det hensigten, at man vil tilgå en liste med alle de udarbejdet videoer der i dokumentationen.

Formålet med Stack Overflow ikonet er, at føre brugeren til Stack Overflow, hvor man vil få en oversigt over de problemstillinger, der er beskrevet i miljøet. Dette er valgt, da brugerundersøgelserne viste, at udviklere både fejlsøger i dokumentationen, samt på Stack Overflow (se figur B.21). Af samme grund, er en sti til GitHub tilføjet. Denne sti giver ydermere mulighed for at tilgå kildekoden, som flere af Karriereudviklere benytter til fejlsøgning.

Den komplette prototype kan ses i bilag D i filmappen “1. version”.

4.2.1 Forside

Som det ses på figur 4.5, er det valgt at indlede forsiden med en forklarende tekst og video.



Figur 4.5. Forsiden

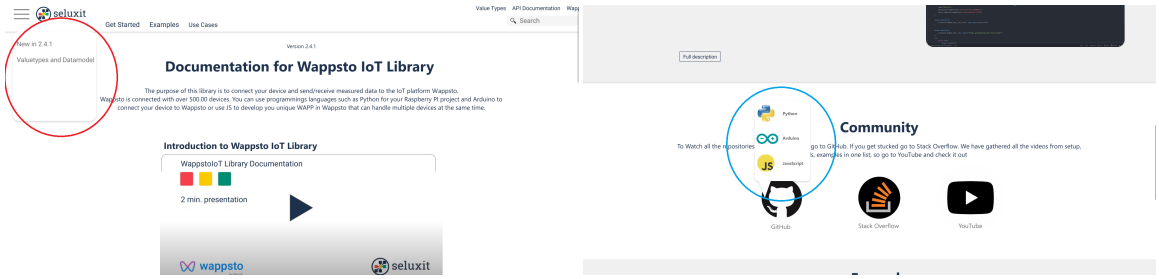
Dette er grundet, at brugerundersøgelserne viste, at det er vigtigt at dokumentationen formidler formålet og brugen af biblioteket (se figur 3.9). Videoen er valgt, da brugerundersøgelserne viste, at flere deltids udviklere fortrak denne kommunikationsform. Dernæst får brugeren mulighed for at tilgå opsætningsguiden via *Get started* knappen. Dette er valgt for at gøre det nemt for brugeren at starte med biblioteket.

Hvis brugeren scroller videre ned vises en Use Case. Her er standardindstillingen sat til

programmeringssproget Python. Det er valgt, at vise en Use Case på forsiden for at kunne understøtte og beskrive mere om biblioteket formål, samt at give inspiration til brugen af det. For at brugeren kan finde en relevant use case, er det valgt at man kan skifte programmeringssprog, hvorved den viste use case vil ændre sig ud fra valget.

Hvis man scroller videre ned fra *Use Cases* sektionen, vil man tilgå sektionen *Community*. Denne sektion har til formål, at give brugeren mulighed for at tilgå kildekoden direkte, samt at se miljøet omkring biblioteket på Stack Overflow. Dette er valgt, da det er en af de parametre, som brugere vælger deres biblioteker på. Til sidst, er der designet for, at man kan tilgå alle videoer i dokumentation via YouTube linket.

Example sektionen har ligesom *Use Cases* sektionen til formål at formidle formålet med biblioteket og hvordan det kan bruges. Til sidst på forsiden, kan man ligeledes tilgå opsætningsguiden via knappen *Get Started*.



(a) *(b)*

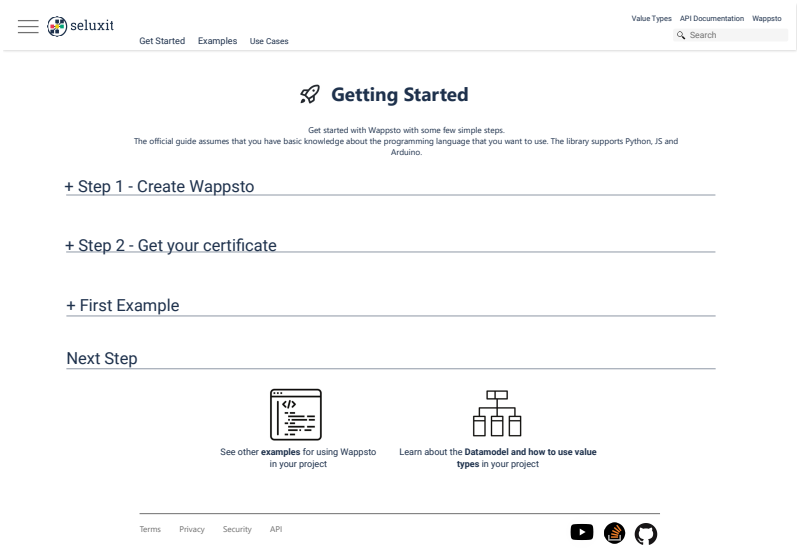
Figur 4.6. Figurene viser forskellige funktioner på forsiden.

På figur 4.6a ses, hvorledes der er designet for burgermenuen øverst til venstre (se rød cirkel). Her er det hensigten, at der skal tilføjes alle hovedsider i dokumentationen for at gøre det lettere for brugeren at navigere rundt på siden. Dette er valgt grundet brugerundersøgelserne viste at nem navigation var et vigtigt parametre ift. indhold i dokumentationen (se figur 3.9).

På figur 4.6b ses den popup menu, der er designet for, hvis brugeren trykker på GitHub ikonet. Denne popup menu har til formål, at navigere brugeren hen til den side, der er relevant for brugeren ift. programmeringssprog. Dette har ligeledes til formål, at gøre navigeringen til det relevante sted nemt for brugeren.

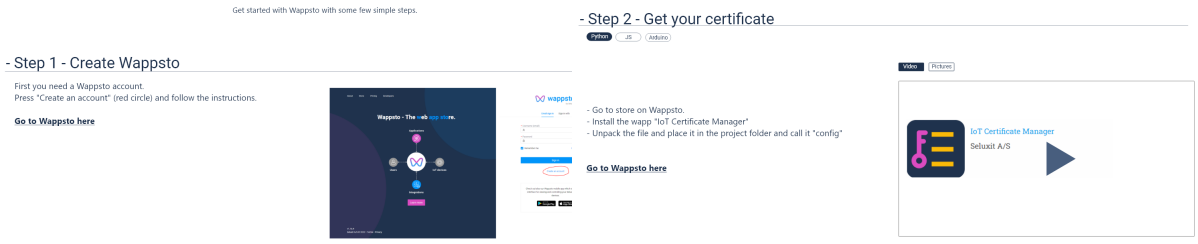
4.2.2 Get Started

På figur 4.7 ses siden for opsætningsguiden. Øverst på siden er der tilføjet en overskrift, samt en introducerende tekst.



Figur 4.7. Working Example

Dernæst er der designet for de to steps opsætningen kræver at fuldføre opsætningen, samt et eksempel. Disse tre punkters information er skjult, så brugeren skal folde hvert step ud for at tilgå informationerne. Dette er blandt andet valgt ud fra resultaterne fra workshoppen (se afsnit 4.1.1), samt for at minimere den kognitive belastning. Ydermere giver dette ekspert brugeren mulighed for kunne tilgå den ønskede information direkte, uden at skulle igennem flere steps.



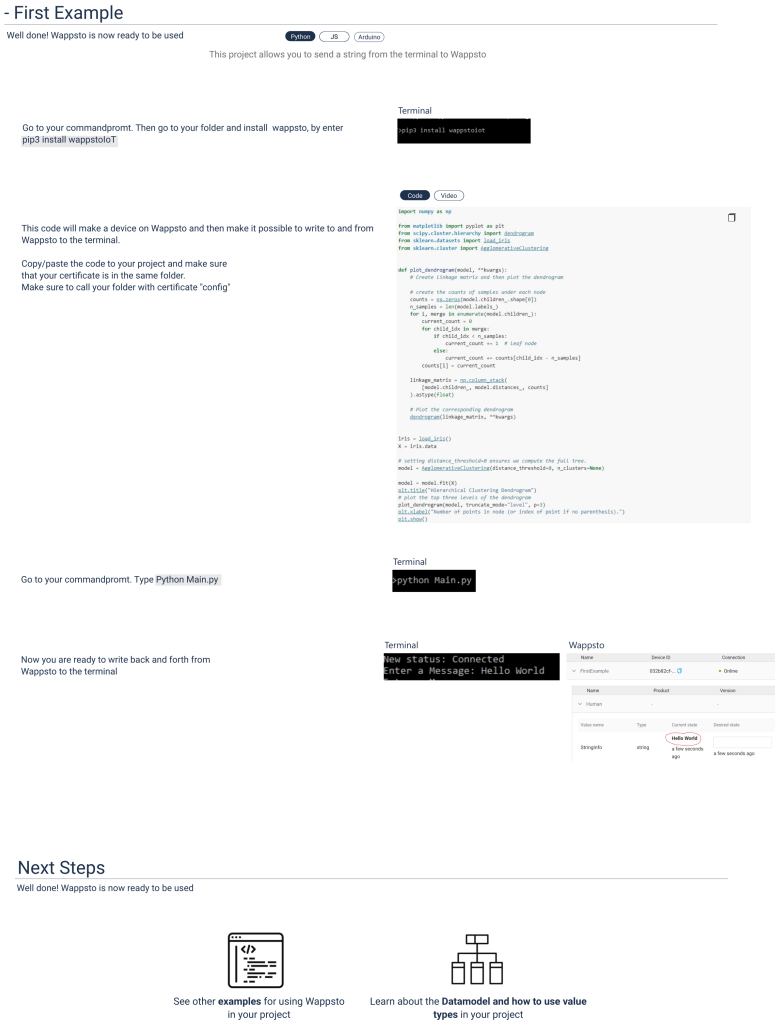
(a) ***(b)***

Figur 4.8. Figurene viser step 1 og 2, når man har foldet hvert step ud.

På figur 4.8a, ses designet for step 1, når brugeren har foldet dette step ud. Her bliver brugeren instrueret i at skulle oprette en Wappsto konto. I steppet er der visualiseret, hvor man skal trykke for at oprette en konto på hjemmesiden. Under den forklarende tekst, er der designet for et hyperlink, der vil navigere brugeren til Wappsto hjemmesiden. Dette har til formål, at lette navigationen til Wappsto for brugeren.

I step 2, som ses på figur 4.8b, bliver brugeren instrueret i at indhente certifikater til deres projekt. Der er designet for, at brugeren har mulighed for at læse den forklarende tekst eller se en video, der gennemgår hele processen. Startsbilledet på videoen viser, hvilken webbaserede applikation, der skal hentes og derfor har til formål at hjælpe de personer, som kun ønsker at læse teksten. Videoen er der dog kun blevet designet for på nuværende stadie af prototypen. At hente certifikater er kun nødvendigt for opsætning til Python og

Arduino, så denne del vil ikke være aktuel for JavaScript opsætningen.

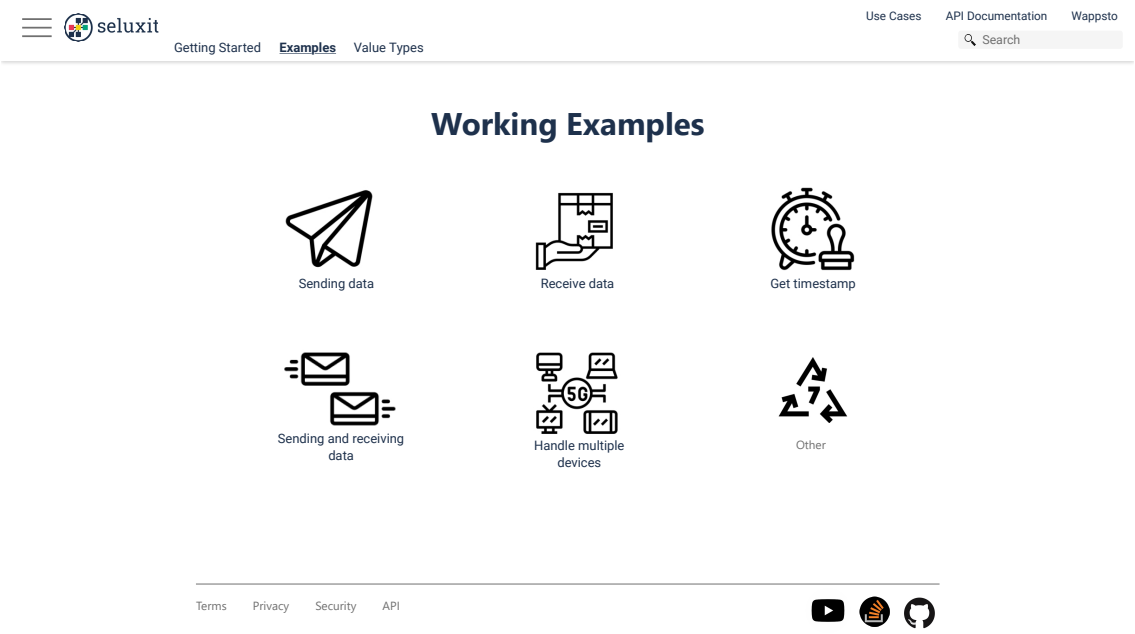


Figur 4.9. Figuren viser punktet *First Example*, når det er foldet ud.

Som det ses på figur 4.9, er der tilføjet et eksempel i opsætningsguiden. Dette eksempel har til formål, at give brugeren en nem start på brugen af biblioteket og eksemplificere for brugeren en måde biblioteket kan anvendes på. Eksemplet er struktureret således, at koden og terminal komandoerne er placeret til højre på siden, hvor den forklarende tekst er placeret til venstre. Til sidst guides brugeren videre til at vælge imellem to ruter. Enten kan brugeren tilgå *working examples* siden eller en side for datamodellen.

4.2.3 Working Example

Dette afsnit har til formål at visualisere ruten for *working example*, samt argumentere for designvalg.



Figur 4.10. Figuren viser steppet før selve eksemplerne, hvor brugeren skal vælge, hvad eksemplet skal omhandle.

Når man som bruger trykker på *Examples* på forsiden, vil man tilgå siden vist på figur 4.10. Denne side har til formål, at give brugeren mulighed for at vælge, hvilket emne eksemplet skal omhandle. Her har ikonerne til hensigt, at gøre dette lettere for brugeren, at genkende de eksempler som man ønsker. Hvis der trykkes på *Sending data*, navigeres man til siden vist på figur 4.11.

Figur 4.11. Working Example

På figur 4.11 ses siden for *Sending data*, hvor der øverst på siden er en tekst, der introducere brugeren til formålet med eksemplet. Dette er valgt, da brugerundersøgelserne viste at generelt formålsbeskrivelse er rangeret højt af både deltidsudviklere og karrriereudviklere (Se figur 3.9). Der er ydermere designet for, at man kan skifte programmeringssprog, da det ønskes at brugeren kun ser indhold, der har direkte relevans for brugerens fortrukne programmeringssprog.

Dernæst vises information til brugeren om, hvad der skal bruges for at gennemgøre dette

eksempel i sektionen "What you need". Efterfølgende er der designet for en sektion kaldet "Introduction to the example", der har til formål, at give en uddybende forklaringen på formålet med eksemplet og på koden selv. Dette forklares via tekst og en video. I den nederst del af siden vil der være en step-by-step guide, hvor siden vil være opdelt i to. Til venstre på siden, er der designet for en forklarende tekst til koden, der beskriver hvorfor man gør som man gør. Til højre er der designet for kode *blocks*, således at brugeren både kan se eksemplet, men også kunne kopiere koden direkte over til sit eget projekt.

Til sidst på siden, er resultatet af eksemplet vist. At vise resultatet af eksemplet, har til hensigt at forstærke forståelsen for formålet for eksemplet.

4.3 Heuristik Evaluering

For at udlede mulige usability problemer i første version af prototypen, samt få indsigt i hvilke designelementer, der bør forbedres på dette stadie af prototypen, vælges det at afholde en heuristik evaluering. Dette afsnit har derfor til formål at dokumentere denne evaluering.

Optagelser, samtykkeerklæring og skitser kan findes i bilag C. Et uddybende referat af evalueringen, samt forsøgsopsætning kan findes i appendiks C

Fremgangsmåde

Fremgangsmåden for den heuristiske evaluering er udarbejdet med inspiration fra Interaction Design Foundation [2017] og Nielsen [1993][s. 155 - 160] fremgangsmåder.

Tre dage forinden selve evalueringen modtog evaluatoren agendaen for evalueringen, samt hvilke heuristikker, der vil blive taget udgangspunkt i. Dette havde til formål at give evaluatoren mulighed for at forberede sig til evalueringen. Evaluatoren fik dog ikke tilsendt prototypen eller tidligere resultater forinden.

Selve evalueringen blev indledt med en introduktion til evalueringen, som indeholdte informationer om hvem målgruppen er, formålet med projektet og agendaen for evalueringen. Efter introduktionen var overstået blev evalueringen opdelt i to faser. I første fase havde evaluatoren til opgave at navigere sig frit rundt i prototypen og forklare sine umiddelbare tanker. I anden fase blev prototypen gennemgået på en mere struktureret måde, hvor evaluatoren havde til opgave at give feedback ud design heuristikkerne udviklet af Jakob Nielsen [1994] .

Under begge faser vil evalueringen forgå som en samtale imellem evaluator og interviewer, hvor der vil blive spurgt ind til løsningsforslag undervejs i evalueringen. Evalueringen blev fortaget med Evelina Mihailova, som er ansvarlig for UX/UI ved Seluxit. Her er hun ansvarlig for web udvikling, grafisk design og information arkitektur. Af uddannelsesbaggrund har Evelina taget en kandidatuddannelsen indenfor *Information Architecture* på Aalborg Universitet, samt en bachelor i grafisk design på UCN. Interaction Design Foundation [2017] forklarer, at en evaluator bør have et indgående kendskab til domænet, samt usability. Grundet Evelinas baggrund ved Seluxit og derved kendskab til domænet, samt uddannelsesbaggrund vurderes det at Evelina netop besidder disse kompetencer og derfor er en ideel evaluator.

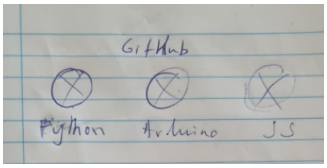
4.3.1 Resultater

Resultaterne er blevet udarbejdet via en manuelt udført videoanalyse, hvor sessionen er blevet gennemgået og resultaterne er blevet noteret ned. I dette afsnit vil der blive fremhævet de mest essentielle resultater fra evalueringen, mens et mere uddybende referat, bliver gennemgået i appendiks C. Evalueringen tog cirka 1 time og 20 minutter.

Evalueringen startede i fase 1 på forsiden af prototypen og her forklarer Evelina indledningsvis, at hun er forvirret over konteksten. Dette er grundet, at Seluxit logoet er blandet ind i et Wappsto bibliotek. Denne forvirring argumentere Evelina for, da Seluxit er virksomheden og wappsto er platformen.

En anden betragtning som Evelina gjorde sig på forsiden var, at man bør kunne vælge programmeringssprog i starten af dokumentationen, så det efterfølgende indhold henvender sig direkte til det pågældende programmeringssprog. Ligeledes bør man altid kunne skifte programmeringssprog uanset hvor man befinder sig i dokumentationen. En anden general betragtning som Evelina gjorde sig, var at *breadcrumps* bør indføres i prototypen for at lette navigation for brugeren.

For at undgå forvirring for brugeren og tydeliggøre, at der er forskel imellem *Examples* og *Use cases* sektionerne på forsiden forklarede Evelina, at man bør designe disse forskelligt. Det blev ligeledes forslået, at man på forsiden burde lave en sektion der erstatter *Community* sektionen, hvor der kun fokuseres på GitHub.



Figur 4.12. Figuren viser evaluatorens skitse af et eksempel på, hvordan man kan designe for links til GitHub.

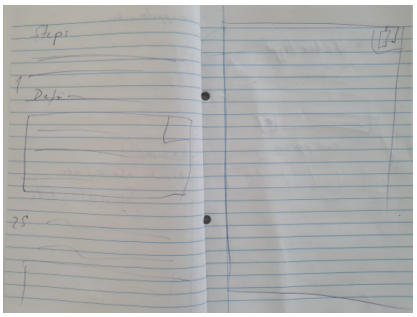
Denne ide bliver eksemplificeret med tegningen, som er vist på figur 4.12. Hertil forklarer Evelina, at man i denne sektion kun kan vælge imellem forskellige programmeringssprog i GitHub. Fravalget af Stack Overflow argumentere Evelina for, at man som bruger sjældent vil tilgå denne platform medmindre man har et konkret problem med biblioteket. Til forsiden forslår Evelina ydermere, at flytte den nederst *Get started* sektion op i toppen af forsiden, for at gøre det nemmere for brugeren at starte med biblioteket.

Evelina forslår ydermere, at man opdeler opsætningen i to sider, således at man designer en side, der kaldes *Get Started* og en side der hedder *Setup*. På denne måde, kan man gøre det nemmere for både erfarne brugere og uerfarne brugere, at lave sin opsætning. Her vil *Get started* have til formål, at introducere biblioteket og dens formål, hvorimod *Setup* siden kun vil skulle forholde sig til selve opsætningen og hvordan den udføres.

Evalueringen af den nuværende *Get started* side viste, at man ifølge Evelina, bør begrænse detaljeringsgraden af informationer. Dette blev argumenteret med, at en bruger i målgruppen ikke vil have behov for et billede af, hvor man opretter en bruger, som det er blevet designet for i step 1 af opsætnings guiden. Derimod bliver det forslået, at man bør

beskrive hvert step med en kort tekst, der forklarer hvad og hvorfor hvert step skal udføres og der efterfølgende bør være links til de steder, hvor det skal gøres.

Ift. *Working example* siden, forklarer Evelina, at når man beskriver en use case, som der bliver gjort i *Sending data* eksemplet, vil man ikke splitte koden op i små sektioner. Dette er grundet brugerens ønske om at kunne kopier koden direkte over i eget projekt og benytte det.



Figur 4.13. Figuren viser evaluatorens eksempel på hvordan man opsætte "Working example" siden.

Som det ses på figur 3.17 foreslår Evelina, at man i stedet opdeler *Working example* siden op i to halve. Den venstre side vil være forklaring af koden og billeder af dele af koden. I højre side, vil den samlet kode være placeret. Dog pointeres det, at man kun bør adskille koden de steder, hvor man kan kopier koden og bruge den direkte uden at skulle kopier fra andre steder. Til den samme side pointere Evelina ligeledes, at man bør være opmærksom på, at stepsene for opsætning og eksempler kan varierer ift. programmeringssprogene. Dette kan have indflydelse for indholdet og antallet af steps. Yderemere bliver det forklaret, at alle billeder af kode bør være HTML blocks, så brugeren nemmere kan kopiere koden over i sit eget projekt.

4.3.2 Diskussion

Dette afsnit har til formål at diskutere de fundne resultater, samt metode brug.

Nielsen [1993][s. 155 - 157] beskriver, at man kun finder ca. 35 % af usability problemerne, når man benytter sig af en evaluator. Derfor kan man argumentere for, at man med fordel kunne have udført denne evaluering med tre til fem personer. Dette vil ifølge Nielsen [1993] gøre, at man vil opdage op imod ca. 75 % af usability problemerne i prototypen. Det har dog ikke været muligt, at rekvirer flere evaluatorer, som både har en indgående forståelse for usability og tredjepartsbiblioteker, som Interaction Design Foundation [2017] fastslår en evaluator bør have. Derfor kan man argumentere for, at en anden metode såsom en traditionel usability test, hvor man får forsøgspersoner fra målgruppen til at udføre specifikke handlinger, vil have været mere fordelagtig på dette stadie af prototypen. Dog vurderes det, at grundet den første fase af evalueringen, hvor prototypen blev gennemgået og antallet af tilgængelige forsøgspersoner i målgruppen er begrænset, har den heuristik evaluering været bedst egnet.

Til sidst i evalueringen udtaler Evelina, at hun ikke anså at prototypen var på et stadie til at udfører en egentlig heuristisk evaluering. Dette er på trods af at Aurora Harley [2018]

nævner at denne metode kan benyttes i alle stadier af prototypen. Det kan derfor diskuteres om evaluerings metoden har været hensigtsmæssigt ift. den pågældende evaluator. Her kunne man i stedet have udført en mere traditionel ekspertgennemgang, hvor der ikke blev taget udgangspunkt i bestemte heuristikker, men hvor det var en mere general gennemgang af prototypen.

Interaction Design Foundation [2017] nævner, at en af ulemperne ved denne evalueringsmetode er, at evaluatoren kan benævne usability problemer som reelt ikke er tilstede. Her kunne man igen med fordel have benyttet flere evaluatorer. Dog vil resultaterne fra evalueringen med Evelina blive opholdt imod tidligere fund og indsamlet viden med det formål at minimere denne ulempe.

4.3.3 Delkonklusion

Formålet med den heuristisk evaluering var, at få indsigt i de større usability problemer på et tidligt stadie af prototypen.

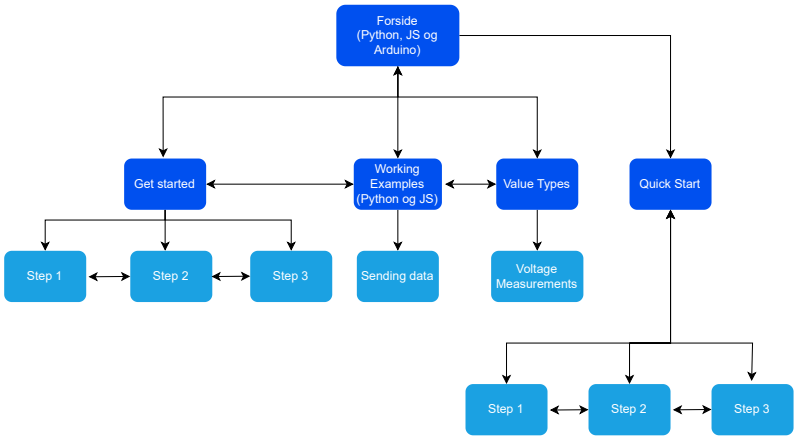
På trods af, at fase 2 af evalueringen ikke blev gennemført efter hensigten, vurderes det stadig at formålet med evalueringen er opnået. Dette er grundet, at fase 1 i evalueringen gav en passende indsigt i de usability problemer, som prototypen har på nuværende stadie. Ud fra evalueringen kan det derfor konkluderes at prototypen skal tilrettes jf. evalueringen.

Tilretningen og argumenterne for disse vil blive præsenteret i næste afsnit.

4.4 2. Version

Dette afsnit har til formål at visualisere den anden version af prototypen, som er blevet redesignet ud fra resultaterne beskrevet i afsnit 4.3. Dette afsnit vil kun præsentere de mest væsentlige ændringer, der er fortaget. Den komplette prototype kan findes i bilag D under "2. version".

Prototypen er stadig udarbejdet i Adobe XD, da dette designværktøj stadig opfylder de krav for prototypen, som der på nuværende stadie.



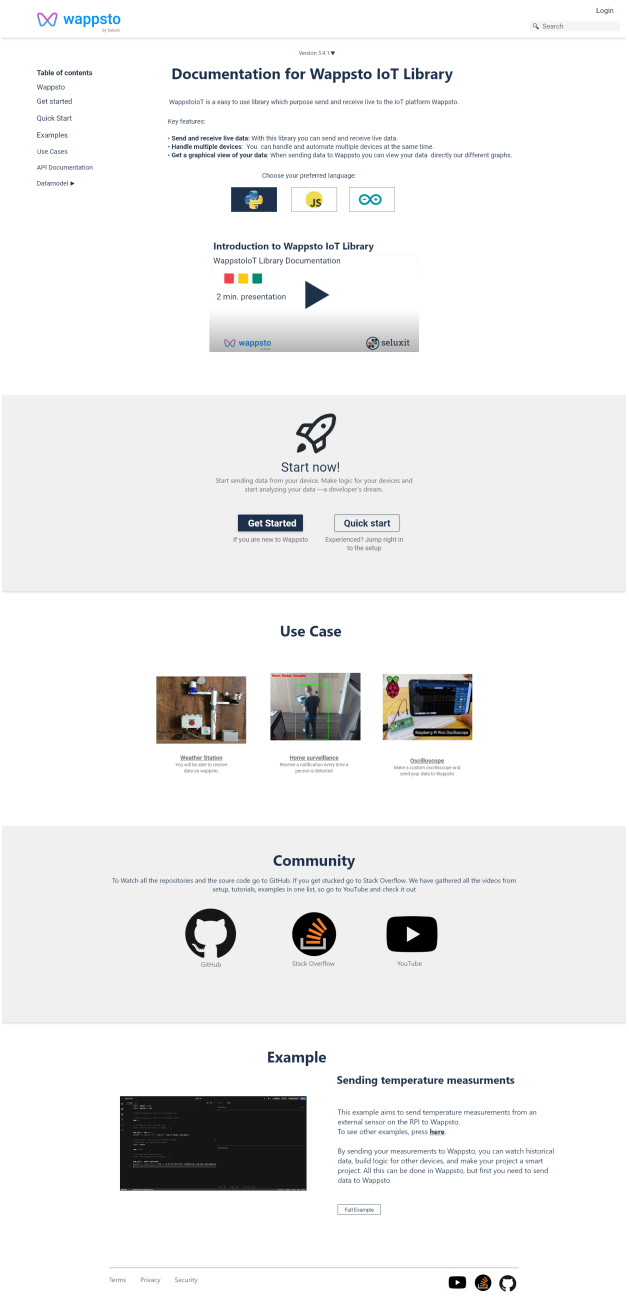
Figur 4.14. Figuren viser flowet for 2. version af prototypen. De steder hvor der tilføjet forskellige programmeringssprog, er der en side for hvert programmeringssprog. Hvis der ikke står noget i kassen udover sidenavnet, er siden kun gældende for Python.

Som det ses på figur 4.14, er der blevet tilføjet flere sider til prototypen ift. første version (se figur 4.4). Der er blandt andet tilføjet en startsside for hvert programmeringssprog, samt en *Working example* side for både Python og JavaScript. Dette afsnit vil tage udgangspunkt i Python.

4.4.1 Forside

Gældende for forsiden i denne version af prototypen er, at det valgte programmeringssprog vil afgøre, hvilke use cases og eksempler, der vises på forsiden. Det er også hensigten, at hvis man trykker på et ikon i *Community* sektionen, så vil man blive navigeret til det relevante sted ift. programmeringssproget.

Ydermere er det tiltænkt at sidemenuen placeret til venstre på siden, skal følge med, når brugeren scroller på siden. Dette har til formål, at gøre det nemmere for brugeren at navigere rundt i dokumentationen.



Figur 4.15. Figuren viser forsiden af prototypen.

Som det ses på figur 4.15, er der fortaget en ændring i strukturen på forsiden ift. første version af prototypen (se figur 4.5).

Forsiden indledes nu med en kortere introduktion, hvor der fremhæves nøglefunktionaliter ved biblioteket. Herefter har brugeren mulighed for at vælge programmeringssprog. Dette bliver efterfulgt af en introduktions video. I første version var *Start now!* sektionen nederst på forsiden. Dog viste den heuristiske evaluering, at denne bør blive flyttet højere op, så brugeren har nemmere ved at komme igang med at bruge biblioteket. Resultaterne fra den heuristiske evaluering (se afsnit 4.3) viste ydermere, at man bør skelne imellem opsætning for en nybegynder og en erfaren bruger. Derfor er der tilføjet siden *Quick Start*, som er

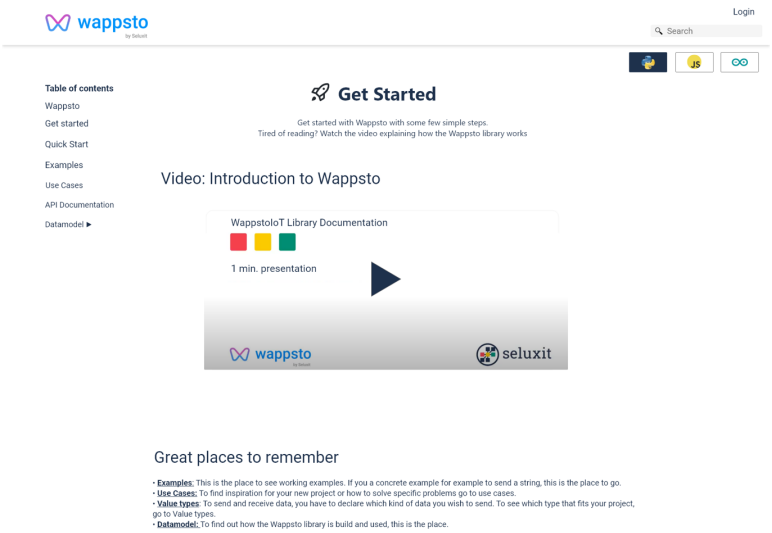
henvendt til den erfarne brugere, hvor *Get Started* siden er henvendt til nybegynderen. En anden ændring på forsiden er, at designet for *Use Cases* sektionen og *Example* er tydeligere adskilt fra hinanden. Dette er valgt, da et af feedback punkterne fra den heuristiske evaluering var, at man kunne forvekslede de to sektioner, da de mindede for meget om hinanden.

Trods feedback om, at sektionen *Community* kun bør indeholde links til GitHub (se figur 4.12), er det valgt at bibeholde det eksisterende design fra første version. Dette er grundet, at man nu vælger programmeringssprog i starten af forsiden og hvert ikon vil derfor navigere brugeren til det relevante sted uden der er behov for en popup menu. Ydermere er det valgt at bibeholde links til Stack Overflow og YouTube, da brugerundersøgelserne viste at brugere generelt fandt disse relevante i både fejlsøgning, men også i valget af bibliotek.

4.4.2 Get Started og Quick Start

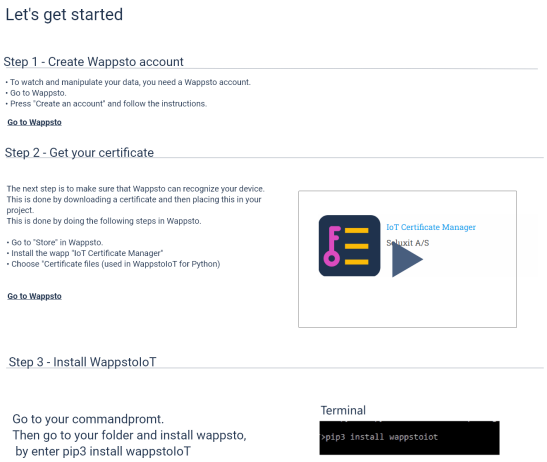
Forklaringen af de større ændringer af *Get started* siden vil være opdelt i to figurer, der skal dog gøres opmærksom på, at i selve prototypen vil disse figurer være samlet på en side. Dette afsnit tager udgangspunkt i opsætning til Python.

Hensigten med *Get Started* siden er, at give den uerfarne bruger en introduktion til biblioteket, hvor blandt andet relevante links og brugsmetoder vil blive fremhævet.



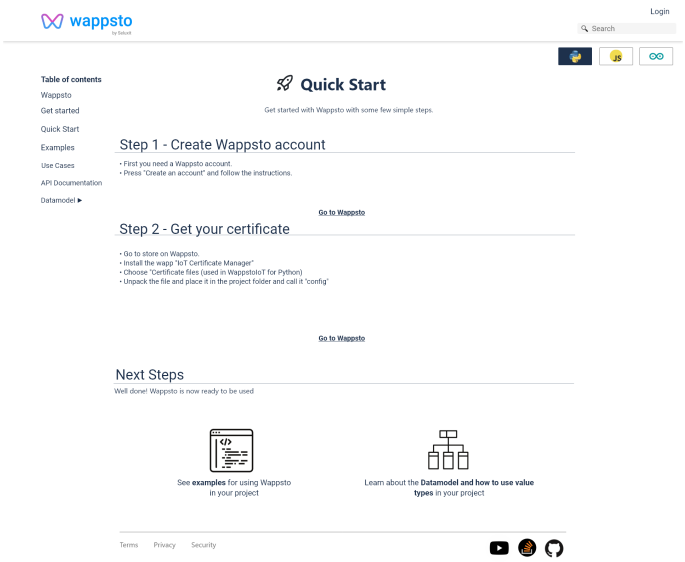
Figur 4.16. Figuren viser den øverste del af *Get started* siden.

Som det ses på figur 4.16, er der øverst på denne side en kort introduktions tekst, som bliver efterfulgt af en video. Denne video har til formål, at introducere brugeren til Wappsto og formålet med WappstoIoT biblioteket. Dernæst er der opsat forskellige links, samt kort beskrivelse med formålet af disse links. Disse links er valgt ud fra en antagelse om, hvilke informationer en uerfaren bruger kan drage nytte af.



Figur 4.17. Figuren viser den midterste del af *Get started* siden.

På figur 4.17 ses det, at informationsniveauet er holdt på samme niveau som i første prototype. Dog er billedet i step 1 fjernet, da evalueringen viste, at dette var overflødigt og virkede forstyrende. Der er ligeledes tilføjet et step mere for Python opsætningen, da man skal installere biblioteket via sin terminal, inden man kan tage biblioteket i brug.

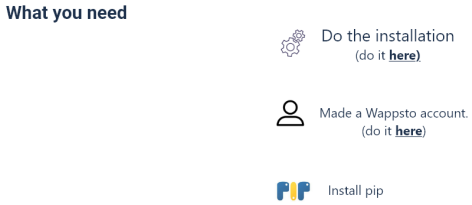


Figur 4.18. Figuren viser *Quick start* guiden for anden version af prototypen.

En tilføjelse til anden version af prototypen, er siden *Quick Start*, som ses på figur 4.18. Denne side har til formål at indeholde den minimale mængde information som opsætningen til biblioteket kræver. Derfor henvender denne side sig til en erfarne brugere, som blot har brug for påmindelse om de forskellige steps til opsætningen. Som det ses på figur 4.14 kan denne side tilgås fra forsiden.

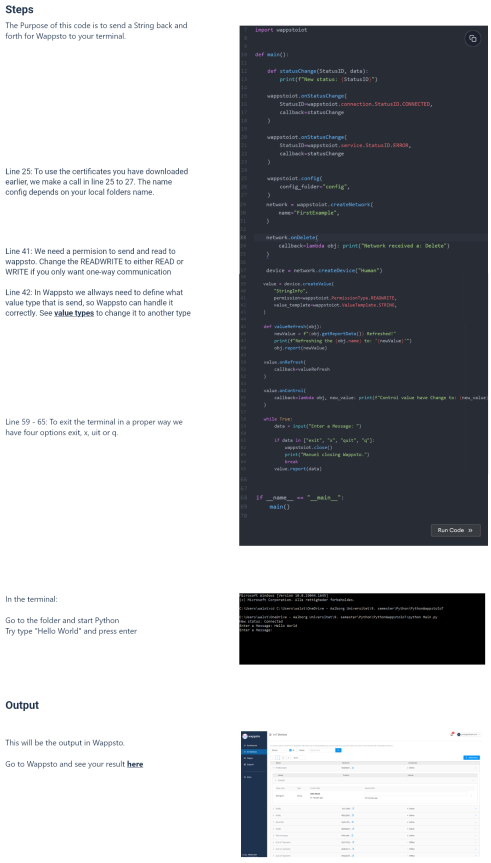
4.4.3 Working Example

På *Working Example* siden, er der kun sket enkelte væsentlige ændringer.



Figur 4.19. Figuren viser sektionen *What you need* på *Working Example* siden. På denne figur vises opsætningen til Python.

En af ændringerne, der er fortaget ift. forrige version er, at ikonerne og teksten i *What you need* sektionen, er flyttet fra at være horisontal placeret til vertikal. Dette kan ses på figur 4.19. Denne ændring er sket på baggrund af resultaterne fra evalueringen, beskrevet i afsnit 4.3.

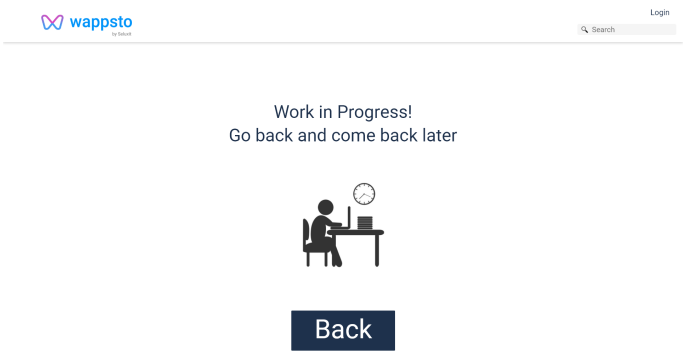


Figur 4.20. Figuren viser den siden som brugerne bliver ført til, hvis de klikker på en rute som der endnu ikke er designet for.

En anden større ændring, ses på figur 4.20. Her ses det, at der i denne version er designet for en html block for hele koden. Dette skal give brugeren mulighed for at kopiere det samlet projekt. Ydermere er der designet for, at man kan kompilere projektet i en online kodeeditor, som f.eks. Codesandbox[2]. Dette er valgt, da teorien gennemgået i kapitel 2

[2] `https://codesandbox.io/?from-app=1`

viste, at brugeren gerne vil kunne se et resultat af koden hurtigt, grundet *active user paradox*.



Figur 4.21. Figuren viser den siden som brugerne bliver ført til, hvis de klikker på en rute som der endnu ikke er designet for.

Stier, som der endnu ikke er designet for, vil ligeledes i version to, gå til siden vist på figur 4.21. Denne side er valgt, at implementere med det formål, at man formidle til brugeren hvilke sider der virker.

4.5 Feedback fra udviklere

En af ulemperne ved den gennemførte evaluering var, at evalueringen kun blev gennemført med en evaluator, hvilket kan true den eksterne validitet. På baggrund af dette vurderes det relevant, at få feedback på 2. version af prototypen fra andre udviklere, således der opnås et mere nuanceret indblik i designet og indhold fra flere forskellige vinkler.

Til dette formål, er det valgt at rekvirerer de samme tre udviklere, som der blev gennemført interviews med på et tidligere stadie i designprocessen (se afsnit 3.2). Disse tre udviklere er valgt, da de alle er domæne eksperter på WappstoIoT biblioteket og alle er karriererudvikler. Dette gør, at de er relevante i dette henseende, da de har et indgående kendskab til WappstoIoT biblioteket og derfor har forståelse for hvilket indhold der essentielt. Yderemere har de alle tre erfaringer med andre tredjeparts biblioteker og kan derfor ligeledes bruge denne erfaring i deres feedback.

Feedback mødet afholdes samlet i en gruppe, da det ønskes at eventuelle kommentarer og holdninger kan blive diskuteret i plenum. Dette har til formål, at minimere sandsynligheden for enkeltstående holdninger opstår og for at sikre forskellige vinkler til spørgsmål om design og indhold. Ydermere er netop disse tre udviklere valgt, da de hver især repræsentere forskellige programmeringssprog og derved kan eventuelle begrænsninger blive belyst under diskussionen.

Fremgangsmåde

Fremgangsmåden for feedback metoden er udarbejdet med inspiration fra Rikke Friis Dam [2021] anbefalet metoder for, hvordan man optimere indsamling af feedback. Mødet vil

foregå i et mødelokale ved Seluxit, hvor der vil være adgang til blandt andet en projektor og whiteboards.

Mødet indledes med en introduktion til formålet for projektet og feedback mødet. Dernæst gennemgås relevante resultater fra de tidligere beskrevet brugerundersøgelser, workshops, evalueringer, samt indsamlet teori. Det vil i introduktionen ligeledes blive gjort klart, at det er en prototype der gennemgås og de således skal være med til at videreudvikle på den og ingen elementer er fastsatte.

Herefter blev det forklaret, at den resterende del af mødet vil bliv delt op i to faser; Præsentation af prototypen og feedback på prototypen. Under præsentation af prototypen vil prototypen blive gennemgået side for side af facilitatoren. Under denne fase vil deltagerne blive instrueret i ikke at give feedback. Dette er for at undgå forvirring over dele af prototypens indhold og derfor ønskes det først at give deltagerne et samlet overblik over indholdet i prototypen, inden der gives feedback.

Under anden fase, vil hver side blive præsenteret igen, hvorefter siden vil blive diskuteret i samlet forum. Når alle sider er gennemgået vil "I Like, I Wish, What If" metoden blive benyttet. Ifølge Rikke Friis Dam [2021] har denne metode netop til formål, at strukturer feedback. Metoden gøre det ydermere muligt for deltagerne, at give åben feedback med udgangspunkt i de tre udsagn. Metoden vil dog blive justeret ift. Rikke Friis Dam [2021] forklaring, således hvert udsagn gennemgås en af gangen og ikke samlet. Dette er for at sikre at alle tre udsagn bliver dækket og fokuseret lige meget på.

Under hele mødet, vil de deltagende blive opfordret til diskussion i plenum med det formål at fremme et nuanceret og gennemarbejdet møde.

Gennemgang af forsøgsopstilling kan findes i appendiks D.

4.5.1 Resultater

Resultaterne er blevet udarbejdet via en videoanalyse, hvor sessionen er blevet gennemgået og resultaterne er blevet noteret.

Feedback sessionen blev udført i et mødelokale ved Seluxit, hvor alle deltagere deltog fysisk. Workshoppen tog ca. 2 timer. Optagelser, samtykkeerklæring og hjælpemidler, som blev brugt til at udføre sessionen, kan findes i bilag E i filmappen "Feedback fra udviklere".

Dette afsnit vil kun præsentere de væsentlige resultater, hvor et mere dybdegående referat bliver gennemgået i appendiks D.

Det første der blev gennemgået var forsiden af prototypen. Her var der en general enighed iblandt udviklerene om, at der på nuværende stadie er for mange informationer på den øverste del af forsiden. Til dette forklarer Michael, at der bør kun stå meget kort, hvad det pågældende bibliotek gør og så skal der være links til en mere uddybende forklaring. Ligeledes blev det forklaret, at der mangler information på forsiden om, hvad man skal bruge programmeringssprogene til. Ud fra dette blev udviklerne enige om, at man bør skifte fokus på forsiden til at man først vælger projekt og dernæst programmeringssprog. Dette blev argumenteret for ved, at man som brugere vil befinde sig i dokumentationen for, at finde ud af om biblioteket løser en specifikt problemstilling. Derfor skal man først

finde ud af om det er tilfældet inden man kan vælge programmeringssprog.

Dernæst blev det forklaret, at der generelt mangler informationer om selve brugen af biblioteket. Hertil forklarede Andreas, at der bør være en dedikeret side til at forklare brugeren, hvordan biblioteket fungerer med f.eks. datamodeller, *value types* og kommunikationen til Wappsto. Andreas tilføjer, ved at forklarer man bør kunne gennemføre eksemplerne uden introduktionssiden uden denne side, men hvis man som bruger ønsker en dybere forståelse for biblioteket, så skal man have muligheden for at tilgå denne side.

Til forsiden blev det ydermere kommenteret, at det er godt at der er use cases og eksempler, så folk kan scrolle ned igennem og få sig en bedre forståelse af biblioteket end hvad den indledende introduktion giver.

Slutteligt om forsiden, blev det forklaret at det er vigtigt, at indholdet på forsiden fanger folks opmærksomhed, således at de hurtigt kan få en forståelse for hvilke muligheder biblioteket giver og om det er relevant for deres problemstilling.

Ift. *Get started* og *Quick start* siderne forklarer deltagerne, at de ikke forstår opdelingen. Ydermere stiller de spørgsmålstegn ved om *Quick start* er hurtigere at benytte end *Get Started* guiden. En af deltagerne forklarer, at han troede at *Quick start* navigerede brugeren til et eksempel, som man kan kopier direkte over i sin kode, hvilket de to andre nikkede genkendende til.

Da samtalen går over på *Sending data* siden, forklarer Karsten, at han vil forvente denne type detaljegrad er gældende for et enkelt eksempel i dokumentationen. Ud fra et overordnet eksempel, skulle man så kunne få en forståelse for de fleste dele af funktionaliteterne i biblioteket. Så når man kommer over i andre eksempler vil man genkende de tidligere funktionaliteter, bare brugt på en anden måde. Her blev det forslået, at man kunne kalde denne side for *how to* guide. Michael konkretisere eksemplet, ved at foreslå at man kunne tage udgangspunkt i et projekt, hvor man kan fjernstyre en termostat på en radiator. Til dette eksempel forklarer Karsten, at det er vigtigt at man forklarer, hvorfor man gør som gør og at det giver brugeren indsigt i hvilke parametre, der kan ændres således det kan relateres til andre projekter. Yderemere blev det forslået, at ud fra denne guide, vil der skulle udarbejdes korte eksempler med relevante brugsscenarier, som f.eks. hvordan man sender en tekst og vælger en farvevælger.

Angående *Value types* siden forklarer både Karsten og Andreas, at disse værdier endnu ikke er blevet implementeret på ordentlig vis, da de ikke dækker alle brugsscenarier og der internt uenighed omkring relevansen af dem.

Slutteligt forklarer både Karsten og Michael, at de godt kan lide forsiden, hvis det øverste del med *key features* bliver ændret. Michael tilføjer ydermere, at han godt kan lide strukturen på *Sending data* siden. Ift. hvad udviklerne ønsker ændret, bliver det forklaret at det primære punkt er konteksten øverst på forsiden. Dernæst forklarer Karsten, at et andet punkt er, at *how to* guiden burde forklarer alle de ting, man kan ændre ved eksemplet. Michael kommer til sidst med et forslag om, at han ønsker at koden bliver highlightet, når man hover over den forklarende tekst i *how to* guiden.

4.5.2 Diskussion

Dette afsnit har til formål at diskutere de fundne resultater, samt metode brug.

Formålet med denne iteration, var at få videreudviklet på resultaterne fra den heuristiske evaluering. Dog kan der argumenteres for, at de tre udviklere ikke har den samme forståelse for, hvordan man opnår høj usability, som tilfældet var med evaluatoren (se afsnit 4.3). Det vurderes dog, at de stadig har kunne give feedback på prototypen ud fra deres erfaringer fra tidligere brugte biblioteker og hvad de har fundet intuitivt at benytte.

Ydermere kan det diskuteres om man har opnået indsigt i holdninger fra hele målgruppen i denne session, da den givet feedback kun kommer fra personer, der både er karriereudviklere og domæneeksperter. Grundet dette kan man diskutere om, det havde været fordelagtigt, at inkludere deltidsudviklere i sessionen, som derved har kunne deltage i diskussionerne ud fra deres erfaringer og holdninger.

Feedback sessionen blev afholdt som et uformelt møde, da alle deltagende i sessionen kender hinanden inklusiv facilitatoren. Den uformelle tilgang havde til formål at skabe et miljø, hvor der frit kunne gives feedback uden at de deltagende skulle være bekymret for reaktionen fra facilitatoren. Dog kan man argumentere for, at denne uformelle tilgang kan have påvirket seriøsisteten i diskussionerne undervejs og at koncentrationen derved også blev påvirket. Det blev dog ikke observeret under videoanalysen, at dette skulle være tilfældet, da samtalen ikke på noget tidspunkt drejede væk fra prototypen og sessionens formål.

Jakob Nielsen [1997] nævner i sin artikel om fokusgrupper, at enhver metode, der er baseret på at spørge brugerne i stedet for at måle eller observere, har den ulempe at de kan producere unøjagtige data. Dette er grundet, at brugerne måske tror, de vil have én ting, selvom de har brug for en anden. På nuværende stadie, er der kun blevet brugt sådanne metoder og det kan derfor diskuteres om, der er indsamlet viden om, hvad brugeren rent faktisk gør og ikke kun, hvad de tror de gør.

Ud fra videoanalysen blev det observeret, at facilitatoren ikke altid forholdt sig neutralt i sine spørgsmål og forklaringer af prototypen, som Rikke Friis Dam [2021] ellers forklarer man bør. Her kan man diskutere om dette har påvirket feedbacken fra de deltagende og der derfor er blevet indsamlet data som er misvisende ift. deltagenens oprindelige holdninger. Dog blev det også observeret i videoanalysen, at der blev givet kritisk feedback på tidspunkter facilitatoren ikke forholdt sig neutralt. Dette kan man argumentere for, at det grundet det uformelle miljø, som har været med til at skabe grobund for dette.

Efter 1 time og 22 minutter, måtte Andreas forlade sessionen. Dette gjorde at de sidste ca. 40 minutter blev gennemført med kun to deltagere. Her kan man diskutere om det har haft indflydelse i diskussionen iblandt de deltagende. Det anses dog til at have minimalt betydning, da den resterende del af sessionen primært omhandlede en opsummering af resultaterne fra den modtaget feedback, samt "I Like, I Wish, What If" metoden.

Feedback sessionen var planlagt til at tage halvanden time, men den varede to timer. Dette gjorde, at de resterende deltagere havde en bagkant, der gjorde at de to sidste punkter i "I Like, I Wish, What If" metoden, blev gennemført overfladisk. Dog resulterede metoden

i en opsummering på resultaterne fra sessionen, og derfor ikke hensigten med konkrete prioriterede emner. Derfor vurderes dette ikke til have større betydning.
En anden konsekvens af den forlængede tid var, at enkelte deltagere udtrykte, at de var trætte i løbet af den sidste halve time af sessionen. Dette kan have været tegn på *fatigue effect*, hvor deltagernes motivation og præstation er faldende. Dog anses dette for kun at være gældende i den sidste halve time af sessionen, da der ellers ikke blev gjort udtryk for dette og alle deltog i diskussionerne på ligefod.

4.5.3 Delkonklusion

Formålet med feedback sessionen var at få et dybere indblik i, hvorledes opbygningen af prototypen bør designes, samt hvilket indhold der bør være på de forskellige sider.

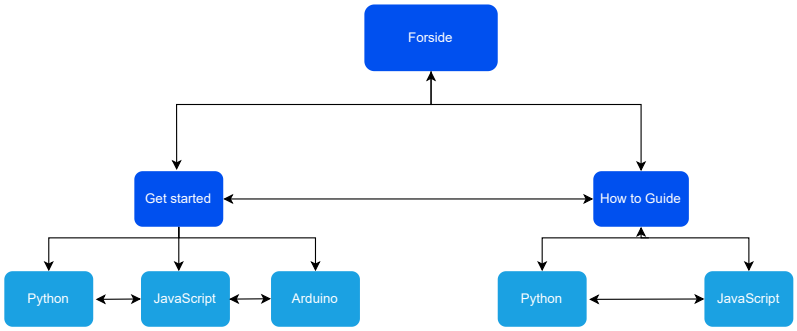
På trods af de begrænsninger, som er blevet diskuteret i afsnit 4.5.2, vurderes det stadig at formålet med sessionen er opnået. Ud fra evalueringen kan det derfor konkluderes, at prototypen skal tilrettes således at der blandt andet opnås en højere forståelse for brugen af biblioteket, samt formålet. Ligeledes skal der fortages ændringer i designet, således brugeren bliver guidet igennem et overordnet eksempel, som forklarer alle væsentlige dele af biblioteket.

Tilretningen og argumenterne for disse, vil blive præsenteret i afsnit 4.6.

4.6 3. Version

Dette afsnit har til formål at visualisere de mest væsentlige ændringer, der er fortaget i tredje version af prototypen. Den komplette prototype, kan findes i bilag D under 3. version. I filen "Links til prototyper", kan tredje version tilgås.

Denne version af prototypen er udarbejdet i Axure[3], da det ønskes at udarbejde videoer til prototypen. Axure giver ligeledes mulighed for at tilføje flere interaktionsmuligheder til prototypen, som f.eks. et funktionelt søgefelt og tilføjelse af HTML kode, samt CSS.
Alle udarbejdede videoer er lavet med dansk tale på nuværende stadie og kan findes i bilag D i mappen "3. version". I appendiks E findes en gennemgang af de nødvendige steps for opsætning, samt gennemgang af kodeeksemplet i *how to* guiden for alle tre programmeringssprog.



Figur 4.22. Figuren viser flowet for 3. version af prototypen.

[3] `https://www.axure.com/`

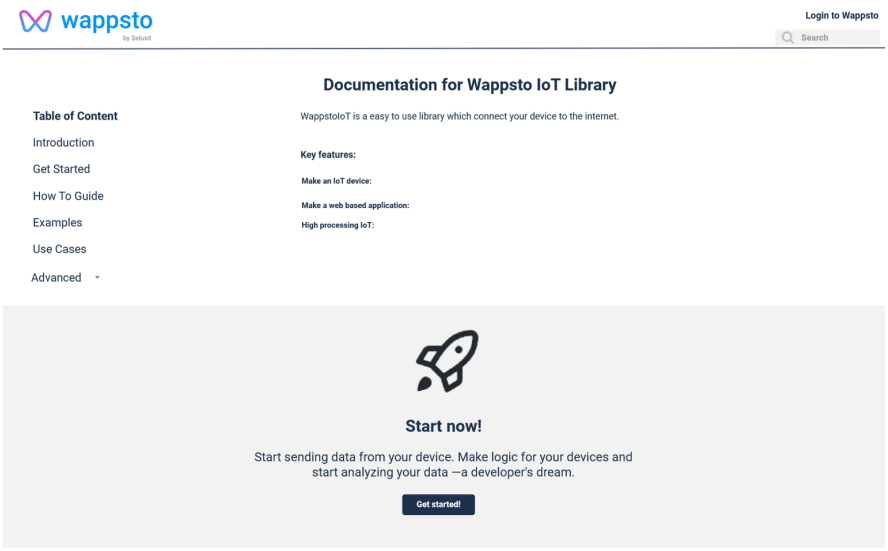
Sammenlignes flowet i anden version af prototypen (se figur 4.14) med flowet fra tredje version (se figur 4.22), ses det at *Quick start* og *Value Types* ikke længere er en del af prototypen.

Quick start siden er blevet fjernet, da udviklerne ikke fandt denne side relevant for opsætningen af biblioteket. Udviklerne forslog, at man i stedet bør dedikere en side til forklaring af formål og brug af biblioteket. Denne vil blive udarbejdet på et senere stadie. *Value types* er blevet fjernet fra prototypen, da dette koncept er unikt for WappstoIoT biblioteket og er ikke er færdigudviklet nok til at indgå i fremtidige test af prototypen.

Ydermere er *Working example* skiftet navn til *How to guide*, da dette blev forslået af udviklerne i afsnit 4.5.1. Derfor er der nu designet for stier til *Examples* og *Use Cases*, hvor *Examples* siden, har til formål at give eksempler på mindre brugsscenarier såsom at sende data. *Use cases* vil være større projekter som f.eks. vejrstationer eller overvågninssystemer. Disse vil dog ikke blive designet for yderligere i prototypen og derfor vil man blive navigeret til siden, som er visualiseret på figur 4.21, hvis der trykkes på disse knapper.

4.6.1 Forside

I redesignet af forsiden, har der grundet resultaterne fra udviklerne været fokus på, at minimere mængden af information, men stadig gøre det muligt for brugeren at opnå en forståelse for formålet med biblioteket.

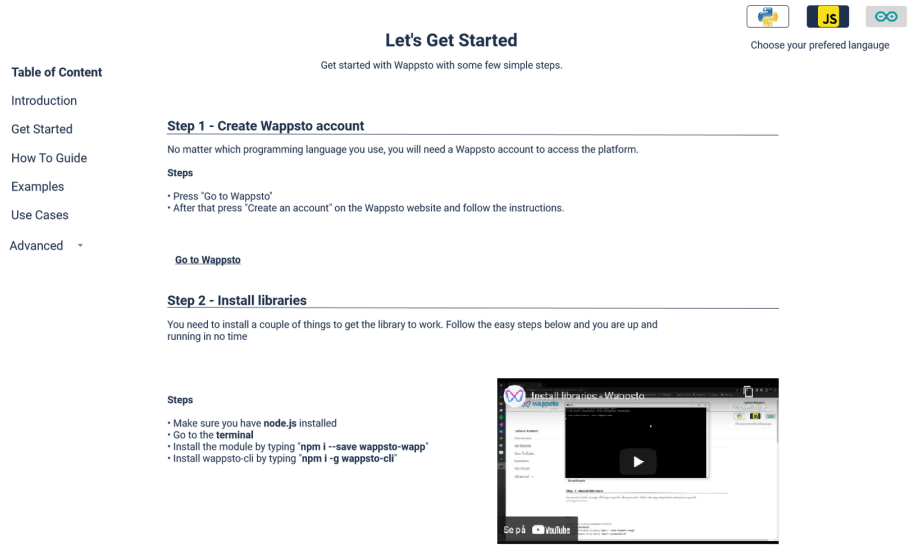


Figur 4.23. Figuren viser den øverste del af forsiden.

Som det ses på figur 4.23, er valg af programmeringssprog og introduktionsvideoen blevet fjernet fra forsiden i tredje version. Ydermere er der blevet tilføjet et punkt i *Table of content*, der hedder *Introduction*. Denne er der dog ikke blevet designet for på nuværende stadie. Ligeledes er *Key features* ændret ift. resultaterne fra afsnit 4.5.

4.6.2 Get Started

Ligesom på forsiden, har fokus i redesignet af *Get started* siden i denne version, været at minimere mængden af information på denne side.



Figur 4.24. Figuren viser den øverste del af *Get started* siden for JavaScript opsætningen.

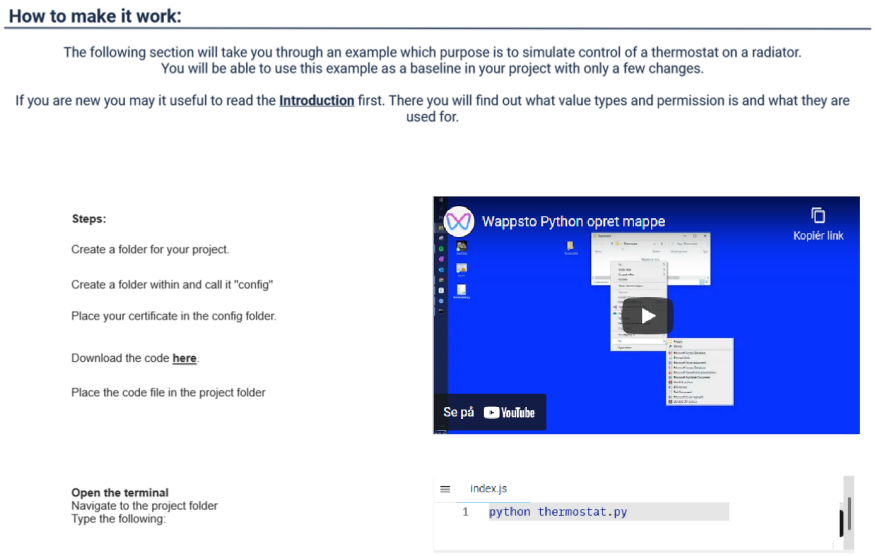
Som det ses på figur 4.24, er det valgt at fjerne introduktionsvidoen fra denne side, samt sektionen med links til relevante steder, når man bruger biblioteket. Dette er valgt for at opnå en kortere opsætningsguide, som kun indeholder relevant information omkring opsætning.

Yderligere ses det på figur 4.24, at der er blevet udarbejdet en video til step 2, der viser hvorledes man installer biblioteket i terminalen. Lignede video er der blevet udarbejdet til Python opsætningen, hvor der vises hvorledes man henter certifikater til sit projekt. Begge videoer kan findes i bilag D i filmappen "3.version".

Nederst på siden, bliver der i denne version linket til *Introduction* siden og *How to* guiden. Dette er en ændring ift. forrige version, da der blev linket til *Value types* og *Examples* siderne. Ydermere er det valgt, at fjerne sektionen med *First example*, da *How to* guiden, har samme formål.

4.6.3 How to Guide

I tredje version har *How to* guiden erstattet *Sending data* siden fra anden version.



Figur 4.25. Figuren viser den øverste del af *Get started* siden for opsætning til Python.

Selvom *How to* guiden er blevet designet med inspiration i *Sending data* siden, er der blevet fortaget nogle ændringer. Der er blandt andet udarbejdet en video for JavaScript projektet, der har til formål at guide brugeren igennem projektet.

Som det ses på figur 4.25, er der placeret forklarende tekst ved siden af videoen. Hensigten med teksten er, at give den mere erfarne bruger mulighed for at skimte sig igennem stepsene, hvor de uerfarne kan blive guidet step for step i videoen.
Der er på nuværende stadie ikke blevet udarbejdet en video til hverken Arduino eller Python.

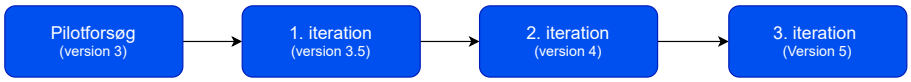
En anden tilføjelse til tredje version, er at der er udviklet et funktionelt kodeeksempel til *How to Guide* siderne for Arduino, Python og JavaScript. Gennemgang af koden, samt datamodellen for WappstoIoT kan findes i appendiks E. I appendiks E findes ligeledes en gennemgang af de trin, som er nødvendige for at få kodeeksemplerne til at fungerer efter hensigten.

Kodeeksemplerne er udarbejdet på baggrund af feedback fra udviklerne i afsnit 4.5. Formålet med disse kodeeksempler er, at danne grundlag for forståelsen af bibliotekets brug. Eksemplet er struktureret således, at de mest essentielle dele af WappstoIoT biblioteket bliver gennemgået. Kodeeksemplerne har alle til formål, at simulere at man kan fjernstyre en termostat på en radiator fra Wappsto. Som der bliver beskrevet i appendiks E, skal der benyttes hardware til Arduino eksemplet, hvilket der ikke skal til de to andre. Grundet denne forskel i sprogene implementeres Arduino ikke på nuværende stadie.

Dataindsamling fra målgruppen 5

Følgende kapitel har til formål at forklare designprocessen for videreudviklingen af prototypen.

Prototypen er indtil nu, kun blevet testet og udarbejdet i samarbejde med karriererudviklere. Som diskuteret i afsnit 4.5.2 nævner Jakob Nielsen [1997], at metoder der er baseret på at spørge brugeren om, hvad de vil have kan producere unøjagtige data. Denne slags metoder er den eneste form, der er brugt indtil videre i designprocessen og derfor har de efterfølgende forsøg til formål, at undersøge hvad brugeren rent faktisk gør. Ydermere har testene til formål, at undersøge om det nuværende design også relatere sig til de personer, som ikke er domæneeksperter. Der vil sekundært være fokus på forståelsen af formålet med biblioteket, da brugerundersøgelserne ligeledes viste, at dette var et vigtigt punkt for både deltidsudviklere og karriererudviklere (se bl.a. figur 3.9). I forsøgene vil der udelukkende være fokus på dokumentationens usability problemer.



Figur 5.1. Figuren viser måden hvorpå prototypen er blevet testet ift. usability forsøg. Versions nummeret under iterations nummeret viser hvilken version, der er blevet testede på.

Som det ses på figur 5.1, er der gennemført usability forsøg over tre iterationer med prototypen. Forinden disse forsøg, er der blevet udført et pilotforsøg.

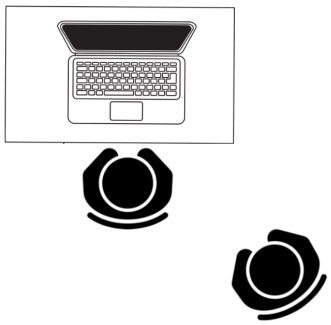
Samtykkeerklæringer, optagelser, udregning af system usability scale (herefter SUS) score og spørgeskemaer for alle versioner af prototypen, kan findes i bilag G. Version 3 af prototypen og frem til version 5 kan tilgås via links, der er placeret i filen "Links til prototyper", som findes i bilag D. I samme bilag, kan videoer og pdf filer ligeledes tilgås.

5.1 Forsøgsdesign

Dette afsnit vil tage udgangspunkt i forsøgsdesignet for første iteration og frem. Dette er grundet, at det gennemførte pilotforsøg udmundende sig i en række ændringer såsom måden hvorpå forsøget skulle optages og introduceres.

Forsøgsopstilling

I udarbejdelsen af forsøgsoptillingen, har der været fokus på at minimere udefrakommende forstyrrelser, samt forstyrrelser fra forsøgslederen.



Figur 5.2. Figuren viser forsøgsopstillingen for usability forsøgene.

Som det ses på figur 5.2, placeres forsøgspersonen ved et bord med egen eller lånt computer. Computeren er tændt og prototypens forside er vist på skærmen. Forsøgslederen placerer sig skråt bagved for ikke at forstyrre forsøgspersonen under forsøget, men samtidig i en vinkel, hvor denne kan observere interaktionen, samt bryde ind, hvis noget uventet skulle ske.

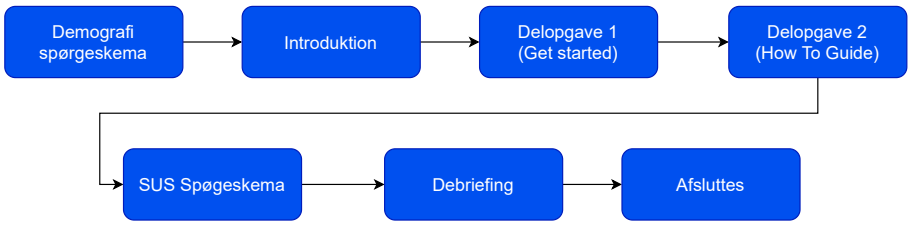
5.1.1 Forsøgspersoner

Rekrutteringen af forsøgspersoner skete ved at kontakte personer i målgruppen, som fandtes i Seluxits netværk og i det personlige netværk. Der blev i alt rekvireret ni forsøgspersoner til forsøgene, hvor der er blevet brugt to til fire forsøgspersoner per iteration. Antallet af forsøgspersoner per iteration er valgt, da Jakob Nielsen [2000] mener, at man via to til fire forsøgspersoner vil finde fra ca. 50 % til ca. 70 % af større *usability*-problemer i ens design. Hvis man benytter sig af flere forsøgspersoner, vil man finde et højere antal usability problemer. Dette er dog blevet fravalgt, da der kun er et begrænset antal forsøgspersoner til rådighed. Ydermere anbefaler Jakob Nielsen [2000], at man bør vægte iterationer over antallet af forsøgspersoner.

Demografiske oplysninger om forsøgspersonerne, samt forsøgspersonernes egne holdninger til erfaringsniveau kan findes i appendiks F.1.

5.1.2 Fremgangsmåde

I dette afsnit vil der blive taget udgangspunkt i det endelige forsøgsdesign, hvor de ændringer, der er fortaget under pilotforsøget, vil blive fremhævet i afsnit 5.2.



Figur 5.3. Figuren viser det overordnet flow for de udførte forsøg.

Som det ses på figur 5.3 skulle forsøgspersonen først udfylde et spørgeskema med demografiske spørgsmål og spørgsmål til erfaring med programmering. Resultaterne fra spørgeskemaet kan findes i appendiks F.1.

Dernæst blev der givet en introduktion til forsøget, som findes i appendiks F.2. Ud fra introduktionen får forsøgspersonen til opgave at udføre delopgave 1, som er at opsætte biblioteket på deres PC via *Get Started* siden i prototypen. Da forsøgspersonen starter på forsiden, har de ligeledes til opgave at finde *Get Started* siden, inden de kan påbegynde opsætningen.

Når opsætningen er færdiggjort, vil introduktionen til anden delopgave påbegynde. I introduktionen vil et nyt scenarie blive forklaret, som ligeledes kan findes i appendiks F.2. Denne opgave vil starte på den side, de sluttede første delopgave med. Her er det hensigten, at de tilgår *How to* guiden og udfører dette eksempel.

Udover at måle metriske data, ønskes det også at undersøge den perciperet usability. Derfor vælges det, at benytte et SUS spørgeskema efter anden delopgave er udført. Til dette spørgeskema bliver forsøgspersonen instrueret i, at de skal fokusere deres svar på dokumentationen og ikke på biblioteket selv. Efter SUS spørgeskemaet er udfyldt, vil forsøgspersonen have mulighed for at komme med uddybende kommentarer i et kommentarfelt. Spørgeskemaet, samt kommentarfelt er udarbejdet i SurveyXact[1]. Efter spørgeskemaet er udfyldt gennemgås en debriefing, hvor forsøgspersonen har mulighed for at komme med feedback på forsøget og prototypen. Efter debriefingen afsluttes forsøget.

I løbet af de tre gennemførte iterationer skete der mindre ændringer i forsøgsdesignet. Heriblandt, at første iteration blev gennemført online over Zoom, men ellers gennemført som beskrevet. Ligeledes blev tredje iteration gennemført ved at forsøgspersonen skulle udfylde et SUS spørgeskema imellem hver delopgave, men ellers som beskrevet.

Ud fra denne fremgangsmåde er det tiltænkt, at den efterfølgende analyse vil bestå af en manuel videoanalyse, notater og udregning af SUS score. For at gøre målingen af SUS scoren meningsfyldt, vælges det at sammenligne den gennemsnitlige SUS score med Jeff Sauro [2018] karaktergivning. Jeff Sauro [2018] forklarer blandt andet, at karakteren A, betyder at systemet har leveret en overlegen præstation, et C er en gennemsnitlig præstation og et D betyder at systemet har dumpet for brugeren.

5.2 Pilotforsøg

Der er blevet udført pilotforsøg med det primære formål, at teste forsøgsdesignet og prototypen for fejl, samt rutinere forsøgslederen. Pilotforsøget blev udført af en forsøgsperson, hvor dennes demografi og ekspertiseniveau, kan ses i tabel F.1 og F.2.

Pilotforsøget tog udgangspunkt i forsøgsdesignet beskrevet i afsnit 5.1. Eneste ændring i forsøgsdesignet var, at det blev optaget via et kamera, placeret ved siden af forsøgspersonen.

5.2.1 Resultater

Resultaterne er blevet analyseret via en eksplorativ videoanalyse, samt gennemgang af noterne fra den efterfølgende debriefing.

Følgende er en opsummering af de overordnet resultater:

[1] `https://surveyxact.dk`

- SUS score: 72,50, hvilket ifølge Jeff Sauro [2018] vil give karakteren C+.
- PFP1 udførte opgaverne i Python.
- PFP1 lagde ikke mærke til, at man kan vælge forskellige programmeringssprog.
- PFP1 udtrykte, at han havde en fin forståelse for opgavernes formål.
- PFP1 udtrykte ønske om, at der generelt skal tilføjes mere kontekst til, hvorfor man skal udføre handlingerne.
- PFP1 fortalte, at lyden på videoerne var lav.
- Grundet vinklen som kameraet optog i, kan skærmen ikke ses.

Følgende er en opsummering af resultaterne for delopgave 1:

- Tid: To minutter og femten sekunder.
- Fejl i dokumentationen, da der mangler en sektion i Python opsætningen, hvor man installere WappstoIoT biblioteket via PIP.
- PFP1 vil fortrække, at der var et link til, hvordan man kan installere PIP.

Følgende er en opsummering af resultaterne for delopgave 2:

- Tid: 18 minutter og 16 sekunder
- PFP1 udtalte, at han meget gerne vil have flere videoer, da det er denne måde han lærer på.
- PFP1 vidste ikke, at denne havde udført opgaven og dette skulle derfor forklares af forsøgslederen.

5.2.2 Diskussion

Pilotforsøget blev udført af kun en person, da antallet af tilgængelige forsøgspersoner er begrænset og ud fra forsøgets formål vurderes dette tilstrækkeligt.

Det lave antal forsøgspersoner gør, at den eksterne validitet er lav og resultaterne kan være unikke for præcis denne forsøgsperson. Ydermere er forsøgspersonen bekendt med Wappsto domænet, da han er ansat ved Seluxit. Dog skal pilotforsøget og resultaterne primært ses som en indledenede øvelse, som har til formål at rutinere forsøgslederen, samt sikre at udførelsen af forsøget kan ske efter hensigten.

5.2.3 Delkonklusion

Forsøget og gennemgangen af pilotforsøget, gav den ønskede indledende indsigt i, hvordan forsøgspersonerne vil interagere med platformen. Hertil blev der fundet elementer i designet, som skal ændres for at øge forståelsen. Ligeledes blev der fundet elementer i prototypen som skal tilføjes, for at forsøgspersonerne kan gennemføre opgaverne.

Følgende liste opsummerer de største ændringer, som er fremkommet i denne version af prototypen:

- Der skal udarbejdes forklarende videoer til dokumentationen både til *Get started* siden og *How to* guiden.
- Det skal tydeliggøres i designet, når eksemplet i *How to* guiden er færdiggjort.
- Der skal generelt tilføjes mere kontekst til, hvorfor man som bruger skal udføre de ønskede handlinger.

- I de fremadrettede forsøg, skal der være skærmoptagelse.

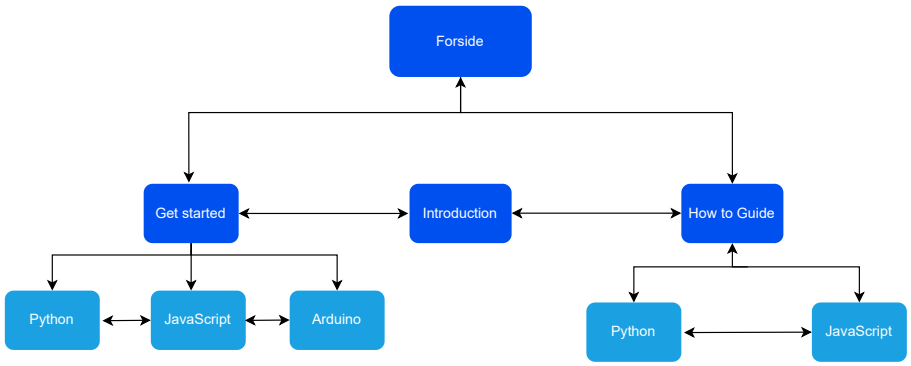
Alle designændringer, der er fortaget på baggrund af evalueringen af Brainstorm prototypen, kan ses i Bilag G i mappen "3.5Version".

5.3 3.5 version

Dette afsnit har til formål at argumenterer for de væsentlige designændringer, der er fortaget på baggrund af pilotforsøgene. Afsnittet har ligeledes til formål, at redegøre for resultaterne fra de forsøg, der er blevet udført på version 3.5.

5.3.1 Ændringer

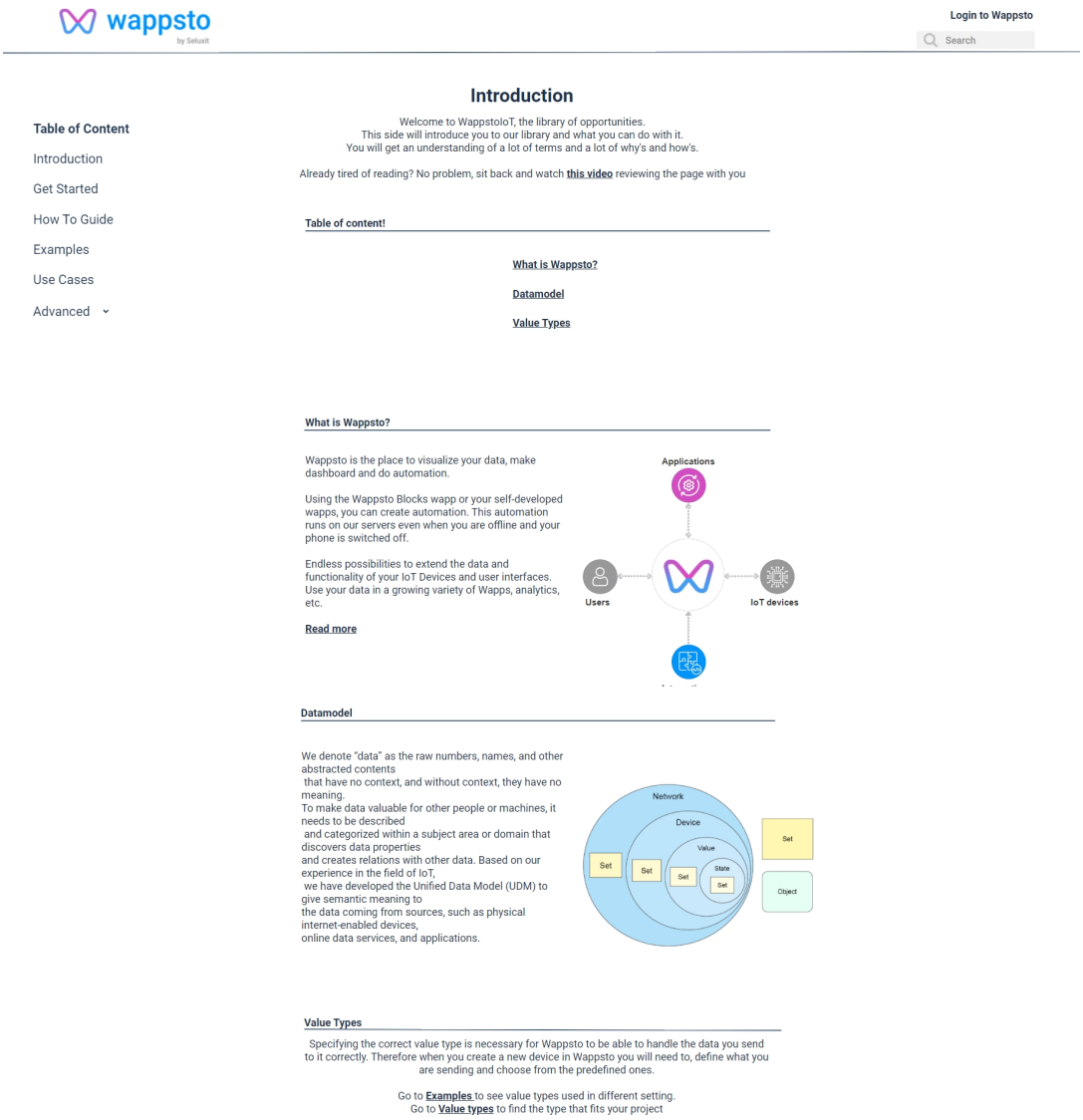
Videoanalysen viste, at man ikke kunne se skærmen i videooptagelsen, og det er er derfor valgt at under de resterende forsøg vil optagelserne forgå via Zoom[2].



Figur 5.4. Figuren viser flowet for version 3.5 af prototypen.

Som det ses på figur 5.4, er der blevet tilføjet en introduktionside til prototypen. Denne side har til formål, at skabe kontekst for Wappsto og give et indblik i, hvordan Wappsto virker.

[2] `https://zoom.us/`

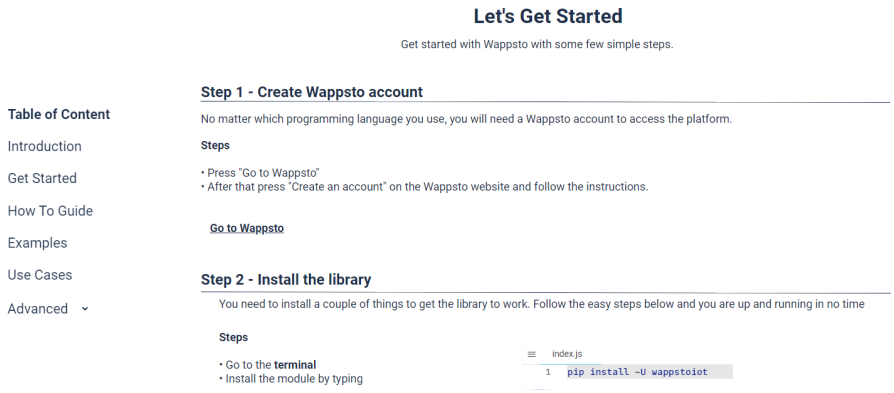


Figur 5.5. Figuren viser siden *Introduction*.

Introduktionssiden som ses på figur 5.5 er designet således, at der først bliver forklaret hvad Wappsto er og hvad det kan bruges til. Dette bliver både gjort illustrativt via en figur og via tekst.

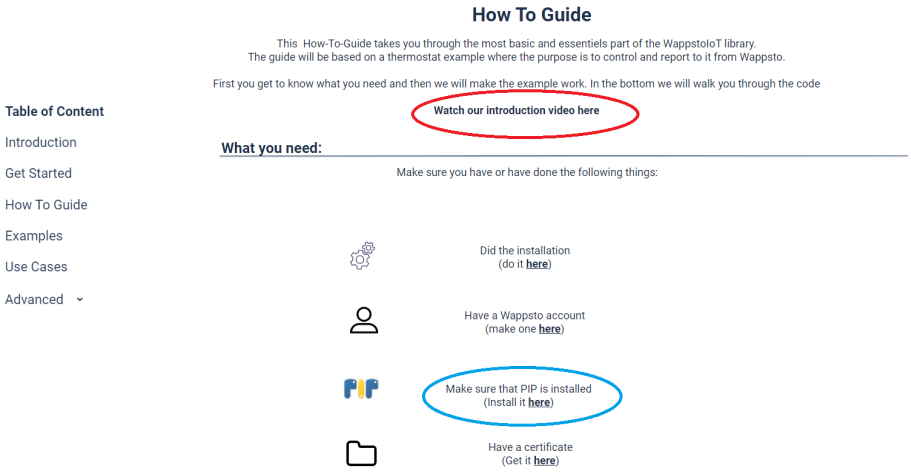
Dernæst vises og forklares datamodellen. Nederst på siden, er der en kort forklaring af *Value Types*. Større dele af teksten og figurene, som bruges til at visualisere forklaringerne er lånt fra Wappsto hjemmesiden[3]. Der er ligeledes designet for en video øverst på siden, som har til formål at gennemgå siden og forklare hvert punkt. Videoen er dog ikke udarbejdet på nuværende stadie.

[3] `https://wappsto.com/`



Figur 5.6. Figuren viser øverste del af *Get started* siden for Python.

Resultaterne fra pilotforsøget viste, at den tidligere version af prototypen ikke guidede brugeren til at installere WappstoIoT biblioteket i terminalen. Derfor er denne handling blevet tilføjet i denne version af prototypen. Som det ses på figur 5.6, er det valgt at denne installation sker i step 2 og brugeren bliver guidet til dette via tekst og visning af den nødvendige terminal kommando.



Figur 5.7. Figuren viser "How to" guiden for Python.

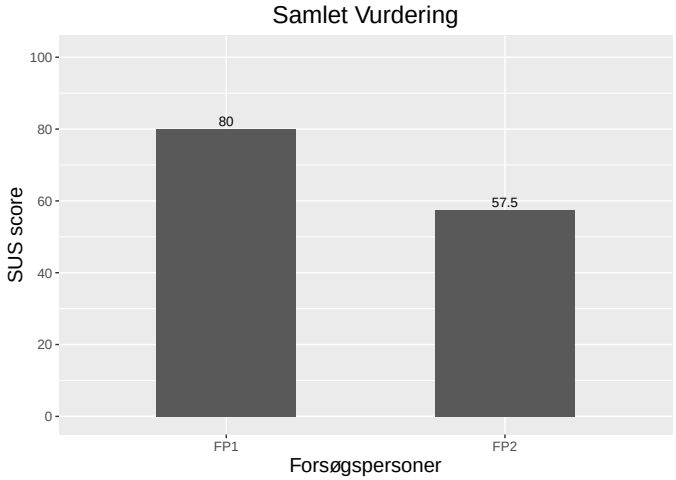
En større tilføjelse til *How to* guiden er, at der er blevet udarbejdet en video, der giver en introduktion til siden. Videoen kan tilgås øverst på siden via et hyperlink (se rød cirkel på figur 5.7)). En anden tilføjelse er et link til guiden for installation af PIP, som ses i den blå cirkel på figur 5.7.

5.3.2 Resultater

Denne iteration af usability forsøg, har til formål at undersøge prototypen for større usability fejl. Forsøget er blevet udført af to forsøgspersoner og deres demografiske oplysninger, samt erfaringsniveau, kan ses i appendiks F.1.

Forsøgsdesignet er som forklaret i afsnit 5.1. Eneste ændringer er, at forsøgene foregik

online grundet større lokation forskelle, samt forsøgspersonerne blev instrueret i at verbalisere sine tanker undervejs i opgaveudførelsen.



Figur 5.8. Histogrammet viser SUS score for forsøgspersonerne. Henad x-aksen vises forsøgspersonerne og opad y-aksen vises SUS-scoren.

Som det ses på figur 5.8, har FP1 vurderet en SUS score på 80, hvilket ifølge Jeff Sauro [2018] vil give karakteren A-. FP2 har derimod vurderet prototypen til en SUS score på 57,5, hvilket giver karakteren D.

FP nummer	Delopgave 1	Delopgave 2	Programmeringssprog
FP1	6 min. og 29 sek.	23 min. og 4 sek.	JavaScript
FP2	7 min. og 57 sek.	48 min. og 7 sek.	JavaScript

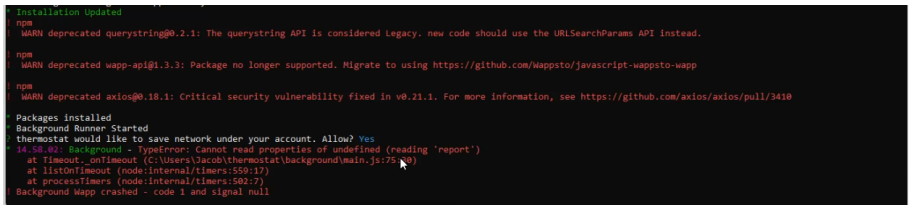
Tabel 5.1. Tabellen viser, hvor lang tid hver forsøgsperson har brugt på de to opgaver, samt hvilket programmeringssprog forsøgspersonerne udførte opgaver i.

Som vist i tabel 5.1 har FP1 brugt kortest tid af de to forsøgspersoner i begge delopgaver. FP2 blev stoppet efter 48 min. og syv sekunder, hvor forsøgsleder og forsøgspersonen sammen fejlsøgte på en fejl, der opstod i terminalen, når forsøgspersonen forsøgte at kompilere projektet.

Følgende liste er kommentarer som forsøgspersonerne har skrevet i kommentarfeltet i spørgeskemaet:

- FP1: Jeg havde lidt problemer med terminalen, men hjælpe-videoerne var en enorm hjælp og gjorde, at man nemt forstod hvordan man skulle løse sin opgave. Så overordnet, hvis man ser bort fra "terminal-besværet", var det nemt at løse sin opgave.
- FP2: constraints.

Den resterende del af afsnittet vil tage udgangspunkt i de observationer der er fortaget, samt noter fra debriefingen.



Figur 5.9. Figuren viser den fejlmedelse som FP2 fik under delopgave 2.

Fejlmeddelelsen vist på figur 5.9 modtog FP2 i delopgave 2, når denne forsøgte at kompilere biblioteket i terminalen. Denne fejl skete 34 minutter og 26 sekunder inde i forsøget. I den resterende del af forsøget, blev der forsøgt at fejlfinde på denne meddelelse. Dette lykkes dog ikke og forsøget blev afsluttet uden FP2 gennemførte denne opgave.

Efter forsøget var overstået viste en nærmere fejlsøgning, at fejlen var opstået grundet brug af et forældet bibliotek.

Figur 5.10. Figuren viser siden *How to guide* i prototypen for JavaScript, ved steppet hvor man skal lave en webbaseret applikation i terminalen.

Brugen af det forældet bibliotek skete ved steppet vist på figur 5.10. Her bliver forsøgspersonen guidet til, at trykke ja til at der skal genereres eksempel kode, hvilket gør at et forældet bibliotek aktiveres.

Følgende liste er en række andre resultater fra forsøgene, baseret på observationer og noter:

- Ingen af forsøgspersonerne var klar over, at de er færdige med opsætningen og skulle gøres opmærksom på dette af forsøgslederen.
- Ingen af forsøgspersonerne lagde mærke til, at man kan vælge forskellige programmeringssprog og spurgte selv indtil om de kunne benytte JavaScript.
- Ingen af forsøgspersonerne benytter sig af videoerne under delopgave 1.
- FP2 forsøger at benytte node.js terminalen under opgaveudførelsen, hvilket ikke er muligt.
- FP2 udtaler flere gange, at han synes videoerne er lange under delopgave 2.
- FP2 laver flere kommando fejl i terminalen og skal have hjælp af forsøgsledere flere gange.
- FP2 udtalte, at han generelt gerne vil have tingene lavet hurtigt i sådanne opgaver og derfor lavede han tit fejl.
- Kun FP1 gennemførte begge delopgaver.

5.3.3 Diskussion

Følgende afsnit har til formål at diskutere de fundne resultater og årsagen til disse.

Der ses en større forskel i den vurderet SUS score for de to forsøgspersoner, hvor FP1 vurdere prototypen til karakteren A-, imens FP2 dumper prototypen med karakteren D. Man kan argumentere for, at grunden til at FP2 dumper prototypen, er grundet at forsøgspersonen ikke gennemførte delopgave 2 og derfor ikke havde samme succesoplevelse som FP1.

FP1 valgte i sin gennemgang af delopgave 2, at se introduktionsvideoen igennem en gang og derefter følge den trin for trin. FP2 gik derimod direkte igang uden at se videoen og lavede op til flere fejl i terminalen. Dette kan tyde på, at FP2 ikke har haft det ønskede niveau til at kunne følge guiden kun på tekst og billeder og ville med fordel have anvendt videoerne. Dette blev yderligere bekræftet, da forsøgslederen guidede FP2 til at se videoerne og derefter kunne FP2 selv gennemføre de sidste trin inden fejlmeddelelsen opstod.
FP2 udtalte i debriefingen, at han så at der var videoer, men følte at han havde travlt med at udføre stepsene i prototypen. Dernæst forklarer han, at det derfor kunne være rart med visse begrænsninger i designet, således at man ikke kunne springe over nogle steps. Dette kan man argumentere for, at dette vidner om "Active user paradox" som forklaret i afsnit 2.2.

Begge forsøgspersoner skulle gøres opmærksom på, at de var færdige med at løse delopgave 1. Derfor kan man argumentere for, at designet ikke har tydeliggjort denne handling nok, så de selv kunne forklare at de var færdige med deres delopgave.

Hvis man kigger i tabel F.2, ses det at begge forsøgspersoner vurdere deres ekspertiseniveau indenfor Wappsto domænet og programmering generelt til at være fra 40 og under. Hertil kan det diskuteres om disse forsøgspersoner er gældende indenfor den ønskede målgruppe, da forsøgspersonerne bliver instrueret i at nul betyder slet ikke erfaren.
Dette kan dog modargumenteres med, at begge forsøgspersoner kunne gennemføre delopgave 1 uden større hjælp og FP1 kunne gennemføre delopgave 2. Ydermere kunne begge forsøgspersoner forklare formålet med brugen af biblioteket i den efterfølgende debriefing.

Ingen af forsøgspersonerne lagde mærke til, at man kunne skifte imellem programmeringssprogene og måtte begge spørge indtil om det var muligt at gennemføre forsøget i JavaScript. Dette vidner om, at knapperne for valget af disse er placeret forkert ift. hvor forsøgspersonen ser. Dette bliver yderligere understøttet, da PFP1 udtalte i afsnit 5.2, at denne heller ikke så man kunne skifte under pilotforsøget.

5.3.4 Delkonklusion

Det vurderes, at forsøget med version 3.5 af prototypen kun delvis gav den ønskede indsigt i de større usability problemer i prototypen. Dette er grundet, at der opstod fejl med dokumentationen undervejs med FP2 i delopgave 2. Dog blev der stadig udledt diverse optimeringsmuligheder for at højne usability niveauet.

Følgende liste opsummerer største ændringer, som er fremkommet i denne iteration.

- Ændring i guiden ift. at sige nej til, at der skal genereres eksempel kode.
- Valg af programmeringssprog skal synliggøres mere.
- Det skal tydeliggøres i designet, hvornår man er færdig med opsætningen.

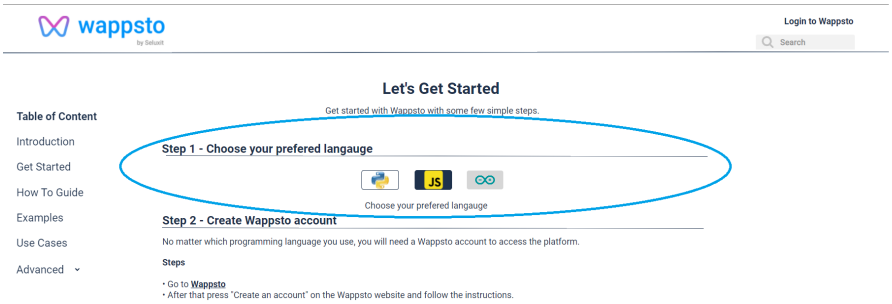
Implementeringen af disse ændringer i prototypen vil blive gennemgået i det følgende afsnit.

5.4 4. version

Dette afsnit har til formål at argumenterer for de væsentlige designændringer, der er fortaget på baggrund af usability forsøgene i den seneste iteration. Afsnittet har ligeledes til formål, at redegøre for resultaterne fra de forsøg, der er blevet udført på 4. version af prototypen.

5.4.1 Ændringer

Sidste iteration af usability forsøg viste, at prototypen har visse fejl og mangler ift. både opsætning og udførelse af *How to* guiden. Derfor er der fortaget ændringer og tilføjelser i designet til 4. version af prototypen.

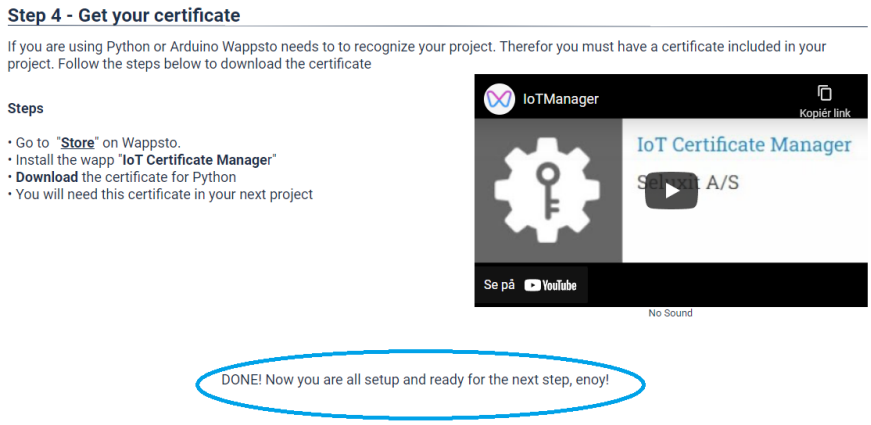


Figur 5.11. Figuren viser siden *Get started* siden i prototypen for JavaScript. Den blå cirkel indikere det tilføjet step, hvilket er gældende for alle tre programmeringssprog.

Resultaterne fra både pilotforsøgene og første iteration af forsøgene viste, at forsøgspersonerne ikke lagde mærke til, at man kan skifte programmeringssprog. Grundet dette, er det valgt at flytte valget af programmeringssprog fra øverst på siden til højre på *Get Started* siden til at være step et i opsætningen, som vist på figur 5.11.

Figur 5.12. Figuren viser siden *How to Guide* siden i prototypen for JavaScript, ved steppet hvor man skal lave en webbaseret applikation i terminalen.

FP2 oplevede i sidste iteration, at der opstod en fejl, da han prøvede at lave en webbaseret applikation (herefter Wapp). Grundet dette, er der som vist på figur 5.12, ændret i teksten ift. version 3.5 af prototypen. Nu fremhæves de vitale oplysninger, som brugeren skal vælge for at få projektet til at virke. Informationer såsom mail adresse og beskrivelse af projektet undlades i denne kontekst for at holde mængden af information på et minimum.



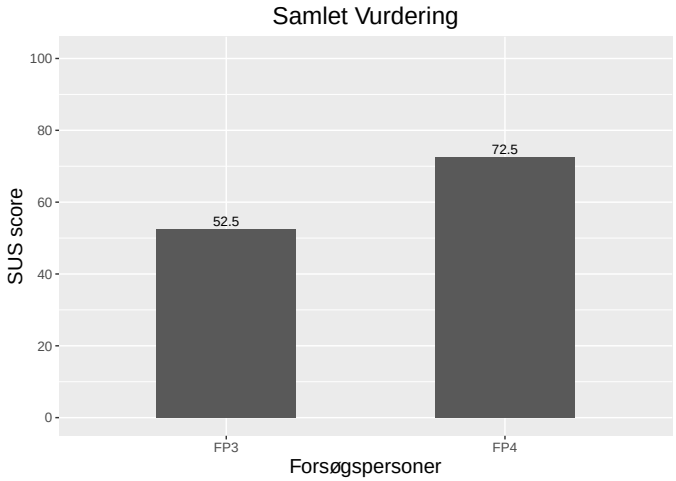
Figur 5.13. Figuren viser slutningen på *Get started* siden for Pythob. Den blå cirkel indikerer den tilføjet tekst ift. sidste version.

Da resultaterne fra sidste iteration viste, at forsøgspersonerne var i tvivl om, de var færdige med opsætningen af biblioteket, er det valgt at tilføje tekst i bunden af opsætningsguiden for både JavaScript og Python. Dette er vist på figur 5.13.

Der er ydermere udarbejdet en ny video, som gennemgår eksemplet i *how to* guiden. Videoen kan findes i bilag D i filmappen "4. Version".

5.4.2 Resultater

Grundet fejlen i tidligere version med opsætningen, er formålet med denne iteration igen at fokusere på usability problemer med opsætningsguiden og *How to* guiden.
Forsøgene er udført som beskrevet i afsnit 5.1.2. Forsøget er blevet udført af to forsøgspersoner og deres demografiske oplysninger, samt erfaringsniveau, kan ses i appendiks F.1. FP3 bruger egen computer til forsøget, hvor FP4 låner en.



Figur 5.14. Histogrammet viser SUS score for forsøgspersonerne. Henad x-aksen vises forsøgspersonerne og opad y-aksen vises SUS-scoren.

Som det ses på figur 5.14, har FP3 vurderet prototypen til en SUS score på 52,5, hvilket ifølge Jeff Sauro [2018] vil give karakteren D. FP4 har derimod vurderet prototypen til en

SUS score på 72,5, hvilket giver karakteren C+. Dette er en forværring ift. sidste iteration, da den gennemsnitlige vurdering for version 3.5 var på 68,75 og denne version har en gennemsnitlig vurdering på 62,5.

FP nummer	Delopgave 1	Delopgave 2	Programmeringssprog
FP3	7 min. og 10 sek.	18 min. og 55 sek.	JavaScript
FP4	7 min. og 29 sek.	11 min. og 46 sek.	Python

Tabel 5.2. Tabellen viser, hvor lang tid hver forsøgsperson har brugt på de to opgaver, samt hvilket programmeringssprog forsøgspersonerne udførte opgaver i.

Som vist i tabel 5.2, var der 19 sekunders forskel i den tid forsøgspersonerne brugte på delopgave 1, hvor FP3 brugte kortest tid. Til delopgave 2, brugte FP3 godt syv minutter længere tid end FP4 på at udføre denne opgave.

Følgende liste er kommentarer, som forsøgspersonerne har skrevet i kommentarfeltet i spørgeskemaet:

- FP3: Very low volume on some videos
- FP4: The only thing that can be a bit confusing is that some might see the device as a combination of multiple sensors and control feedback

Den resterende del af afsnittet vil tage udgangspunkt i de observationer der er fortaget, samt noter fra debriefingen.

Figur 5.15. Figuren viser siden *How to guide* i prototypen for JavaScript, ved steppet hvor man skal kopiere kode over i en main.js fil.

Under udførelsen af delopgave 2, kopier FP3 kun de synlige linjer fra koden vist på figur 5.15. FP3 erstatter herefter den eksisterende kode i main.js filen med disse seks linjer kode og forsøger at kompilere projektet. Forsøgslederen hjælper herefter forsøgspersonen med at få kopieret resten af projektet over.

Følgende liste er en række andre resultater fra forsøgene, baseret på observationer og noter:

- FP3 skulle guides væk fra at bruge node.js terminalen.
- FP3 forklarer i debriefingen, at han ikke følte sig motiveret til at se videoerne efter at have set introduktionvideo i *How to* guiden for JavaScript. Dette var grundet, at den var lang og han følte, at han ikke fik relevant information fra den.
- FP3 klager over lav lyd i introduktion videoen til JavaScript i *How to* guiden.
- FP3 er ikke klar over, hvilke program der kan benyttes til atl åbne main.js filen. Forsøgslederen forklarer, at han kan bruge Visuel Studio, som forsøgspersonen har installeret.

- FP3 oplevede en fejlmeddelelse under kommandoen *npx wappsto serve*, som forsøgslederen skulle afhjælpe.
- FP4 vil først udføre opgaven i Arduino, men får via prototypen feedback på at dette endnu ikke er muligt.
- Da FP4 låner en computer med Windows styresystem, skal forsøgspersonen have hjælp til at finde terminalen, da forsøgspersonen er vant til Linux styresystem.
- FP4 læser den øverste del af *How to* guiden for Python og forsætter derefter et stykke tid med at prøve sig frem i stedet for at følge dokumentationen.
- Lyden fra FP4 bliver forvrænget i den sidste halvdel af forsøget.

5.4.3 Diskussion

Følgende afsnit har til formål at diskutere de fundne resultater og årsagen til disse.

Resultaterne fra det demografiske spørgeskema, som er vist i tabel F.2, viste at FP3 vurdere sin egne programmeringsevner til 80 ud af 100, hvor 100 er meget erfaren. FP4 vurdere sine evner til 46. Ligeledes vurdere FP3 prototypen til karakteren D, hvor FP4 vurdere prototypen til C+. Ud fra denne forskel kan det diskuteres om prototypen på nuværende stadie, henvender sig mere til den uerfarne udvikler end den erfarne. Dog er resultaterne af spørgeskemaet en subjektiv vurdering af egne evner og forudsætningerne for deres bedømmelse af egne programmeringsevner, kan have været baseret på forskellige grundlag. Til dette blev det observeret, at FP3 havde flere problemer med kommandoer i terminalen, som f.eks. hvordan man navigere i mapperne og skulle søge hjælp til dette. Ydermere valgte FP3 at kopier seks linjer kode over i main.js filen, som vist på figur 5.15. Dette kan vidne om en manglende forståelse for opgaven, da disse linjer kode kun har til formål at initialisere WappstoIoT biblioteket og en opdateringsinterval. Dette vidner derfor om en generel forståelse for bibliotekets formål og brug.

Da forsøgspersonerne først skal udfylde et SUS spørgeskema efter at have endt begge delopgaver, vides det ikke hvilken delopgave, der har haft størst indflydelse på vurderingen. F.eks. oplevede FP3 en fejlmeddelelse i terminalen til sidst i delopgave 2. Dette kan man diskutere om det kan have påvirket FP3's overordnet indtryk af dokumentationen i en signifikant grad. Ydermere kan oplevelsen med andre platforme end dokumentationen også have influeret i bedømmelsen, selvom forsøgspersonerne bliver instrueret i at kun at forholde sig til dokumentationen.

Videoanalysen viste, at FP4's tilgang til opgaverne var, at prøve sig frem i stedet for at følge dokumentationen. Dette kan man argumentere for, at dette vidner om "Active user paradox", som forklaret i afsnit 2.2 og som også var tilfældet for FP1.
FP4 vendte dog tilbage til dokumentationen efter, at have prøvet nogle forskellige ting, som ikke virkede. Derfor kan man argumenterer for, at dokumentationen stadig virker efter hensigten, da brugeren kan skabe sig et overblik og prøve nogle ting af og vende tilbage til dokumentationen, hvis det ikke virkede.

Under videoanalysen af FP4's gennemgang, blev det ligeledes opdaget at lyden blev forvrænget i den anden halvdel af forsøget. Dette var grundet, at forsøgspersonen valgte at bruge høretelefoner for at se introduktionsvideoen for Python. Derfor kan man argumentere for, at dette har medført tab af data ift. den kommunikation forsøgsperson og forsøgsleder

har haft, samt fra debriefingen. Dog kan det antages, at større usability problemer vil blive noteret undervejs af forsøgslederen og derfor vil væsentlige resultater ikke gå tabt.

5.4.4 Delkonklusion

Forsøget med 4. version af prototypen gav den ønskede indsigt i de større usability problemer i prototypen.

Følgende liste opsummerer de største ændringer, som er fremkommet i denne iteration.

- Ændring i forsøgsdesignet således forsøgspersonerne udfylder et spørgeskema efter endt delopgave.
- Det skal tydeliggøres, at det er hele koden i *how to* guiden, der skal kopieres over i main.js filen.
- Der skal udarbejdes nye videoer, der er kortere.
- Videoerne skal ligeledes indtales på engelsk.
- Gør opmærksom på i prototypen, at der vil blive taget udgangspunkt i Windows styresystem.

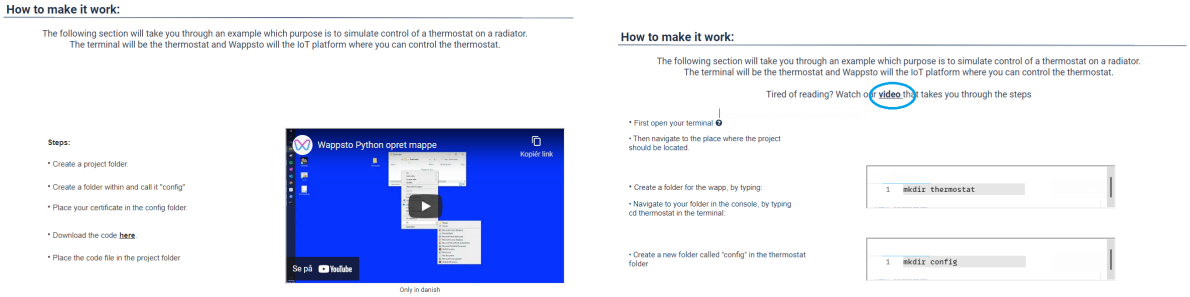
Implementeringen af disse ændringer i prototypen vil blive gennemgået i det efterfølgende afsnit.

5.5 5. Version

Dette afsnit har blandt andet til formål, at redegøre for resultaterne fra de forsøg, der er blevet udført på 5. version af prototypen.

5.5.1 Ændringer

Til denne iteration er der lavet en ændring i forsøgsdesignet. Når forsøgspersonen er færdig med delopgave 1, vil de få til opgave at udfylde et SUS spørgeskema og der vil blive fortaget en kort debriefing. De vil stadig skulle udfylde et spørgeskema efter delopgave 2. Til hvert spørgeskema bliver forsøgspersonerne instrueret i, at de kun skal forholde sig til prototypen og den opgave de lige har udført.



(a) 4. version ***(b)*** 5. version

Figur 5.16. Figurene viser to forskellige versioner af *How to make it work* steppet i *How to Guiden* for Javascript.

En anden større ændring, der er fortaget i denne version er, at videoen som ses på figur 5.16a, er redesignet til at være et hyperlink, som ses i den blå cirkel på figur 5.16b.
Dette redesign er sket for at gøre for at gøre plads til terminal kommandoerne, som ses på figur 5.16b. Redesignet er sket på baggrund af resultaterne fra både version 3.5 og 4, der viste at flere af forsøgspersonerne havde svært ved at udføre opgaverne i terminalen og derfor har brug for en mere detaljeret guiden.

En tredje større ændring er, at der nu er udarbejdet engelsktalende videoer til alle link i opsætning og *how to* guiden, for både Python og JavaScript. I disse videoer har der været særlig fokus på, at holde dem så korte som muligt, da flere forsøgspersoner har udtalt en manglende motivation for at se videoerne, hvis de har været lange.

5.5.2 Resultater

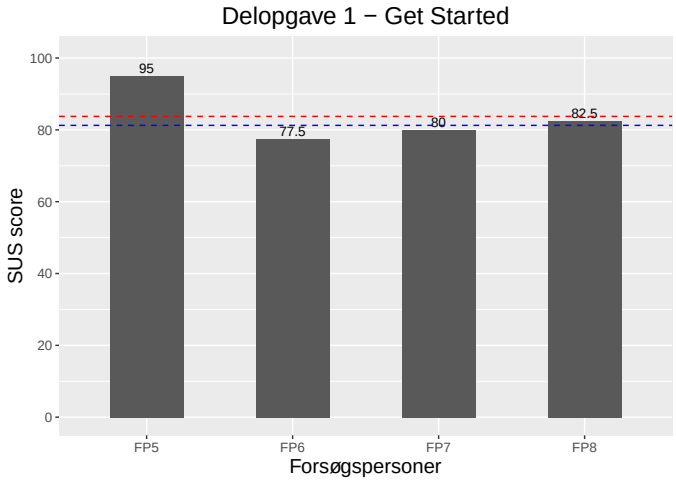
Formålet med denne iteration, er som tidligere at udlede større usability problemer med opsætningsguiden og *how to* guiden.
Det vurderes, at prototypen er på et stadie nu, hvor der kan gennemføres usability forsøg med flere end de hidtidige to personer. Ved at benytte fire forsøgspersoner finder man ifølge Jakob Nielsen [2000] ca. 70% af de usability problemerne prototypen måtte have.

Forsøgene er udført som beskrevet i afsnit 5.1.2. Forsøgspersonernes demografiske oplysninger, samt erfaringsniveau, kan ses i appendiks F.1.

FP nummer	Delopgave 1	Delopgave 2	Programmeringssprog
FP5	7 min. og 23 sek.	11 min. og 28 sek.	Python
FP6	6 min. og 2 sek.	11 min. og 20 sek.	Python
FP7	4 min. og 6 sek.	14 min. 27 sek.	Python
FP8	4 min. 28 sek.	7 min. 41 sek.	Python

Tabel 5.3. Tabellen viser, hvor lang tid hver forsøgsperson har brugt på de to opgaver, samt hvilket programmeringssprog forsøgspersonerne udførte opgaver i.

Som det ses i tabel 5.3, er forskellen på den hurtigste gennemførelsen af delopgave 1 til den langsomste 3 minutter og 17 sekunder. Ligeledes ses det, at FP8, som havde den hurtigste gennemførelse var 1 minut og 51 sekunder hurtigere end den langsomste gennemgang, som var FP5. Ydermere vises det på tabellen, at alle fire forsøgspersoner gennemførte opgaverne i Python.



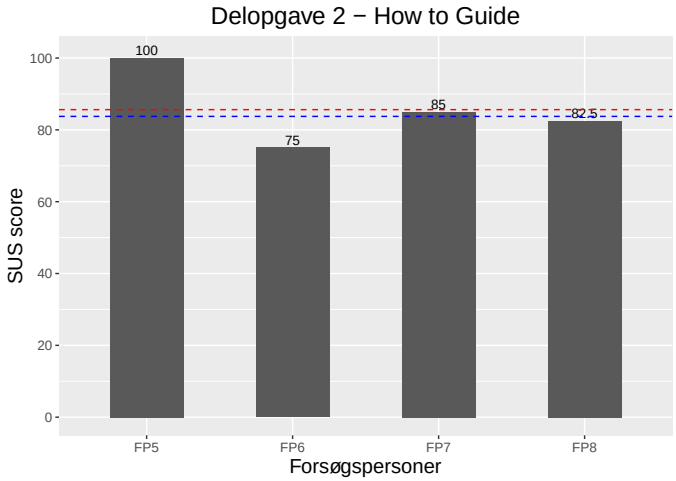
Figur 5.17. Histogrammet viser SUS score for delopgave 1. Henad x-aksen vises forsøgspersonerne og opad y-aksen vises SUS-scoren. Den blå horisontale linje er medianen og den røde horisontale linje er gennemsnittet.

På figur 5.17 ses det, at FP6 vurderer opsætningsguiden, som den laveste af de fire forsøgspersoner. FP6 har vurderet prototypen til en SUS score till 77,5, hvilket ifølge Jeff Sauro [2018] vil give karakteren B+. De resterende forsøgspersoner har vurderet opsætningsguiden til karakterne A-, A og A+.

På figur 5.17 ses det ligeledes, at den gennemsnitlige SUS score er 83,75 og medianen er 81,25, hvilket ifølge Jeff Sauro [2018] begge vil give karakteren A.

Følgende liste er kommentarer, som forsøgspersonerne har skrevet i kommentarfeltet i spørgeskemaet, efter at have færdiggjort delopgave 1:

- FP5: Maybe make the pip install line more noticible that it can be copy-paste
- FP6: I think that with an addition of the language name below the logos, it would be easier for beginners to recognise the different languages.
- FP7: Did what i would expect. nothing unexpected in the process - intuitive
- FP8: Nothing to note



Figur 5.18. Histogrammet viser SUS score for delopgave 2. Henad x-aksen vises forsøgspersonerne og opad y-aksen vises SUS-scoren. Den blå horisontale linje er medianen og den røde horisontale linje er gennemsnittet.

På figur 5.18 ses det, at FP6 har vurderet *How to* guiden som det laveste af de fire forsøgspersoner. FP6 har vurderet prototypen til en SUS score på 75, hvilket ifølge Jeff Sauro [2018] vil give karakteren B. De resterende forsøgspersoner, har vurderet *How to* guiden til karakterne A og A+.
På figur 5.18 ses det ligeledes, at den gennemsnitlige SUS score er 85,63, hvilket ifølge Jeff Sauro [2018] vil give karakteren A+. Medianen er 83,75, hvilket vil give karakteren A.

Følgende liste er kommentarer som forsøgspersonerne har skrevet i kommentarfeltet i spørgeskemaet, efter at have færdiggjort delopgave 2:

- FP5: Seems a good guide
- FP6: Nope
- FP7: Ingen bemærkninger. - rettelse : python3 [filename].py
- FP8: Nothing to add

Følgende liste er en række andre resultater fra forsøgene, baseret på observationer og noter:

- Under FP6 gennemgang af delopgave 2, oplevede denne flere problemer med prototypen, som f.eks. at link til *Introduction* siden ikke virkede og links til videoer.
- FP6 udtalte under debriefing, at forsøgspersonen ikke er bekendt med udtrykket "terminal".
- FP6 udtalte under debriefing, at forsøgspersonen ikke er vant til at arbejde i terminalen og derfor havde svært ved denne del af opgaven.
- FP6 forklarer under debriefing, at forsøgspersonen havde set, at der var videoer, men at han tænkte at han vil se dem i en nødsituation. Dette bliver uddybet med, at han er typen, der bare gerne vil have færdiggjort det her og så komme videre
- Lyden forsvinder i starten af FP5's gennemgang
- Lyden er skrattende for videooptagelsen af FP7 gennemgang
- Der er ikke blevet optaget lyd ved FP8's gennemgang.
- Alle forsøgspersoner udtalt i debriefingen, at de har fået en forståelse for bibliotekets formål.

- Ingen af forsøgspersonerne brugte *Introduction* siden under delopgave 1.
- To af forsøgspersonerne benytter *Introduction* siden i starten af delopgave 2
- FP7 forsøger først at trykke sig ind på *Use cases* og *Examples* i starten af delopgave 2.

5.5.3 Diskussion

Følgende afsnit har til formål at diskutere de fundne resultater og årsagen til disse.

Da det er i projektets interesse at undersøge forståelsen af biblioteket kan det diskuteres om det havde været fordelagtigt, at tilføje en delopgave til forsøget, der havde til formål at undersøge netop dette. Dog har forsøgstiden allerede på nuværende tidspunkt været på omkring en halv time og en tilføjelse af delopgave vil derfor højne risikoen for *fatigue effect*. Ud fra resultaterne fra denne iteration, kan det dog disktures om opsætningsguiden og *How to* guiden nu er på et stadie, hvor der bør fokuseres mere på forståelsen af biblioteket.

FP6 var den forsøgsperson der vurderede prototypen lavest både for delopgave 1 og 2. FP6 var også den eneste der oplevede problemer med prototypen bl.a. hvor FP6 forsøgte at finde informationer omkring, hvad en terminal var. Her virkede prototypen ikke efter hensigten og forsøgspersonen fik ikke den hjælp, som der ellers var designet for. Det kan derfor diskuteres om dette har haft en medvirkedene faktor til den lave vurdering.

Som det kan ses i tabel F.2 er FP6 dog også den forsøgsperson, der har vurderet egen erfaring med programmering, som den laveste af de fire forsøgspersoner. Her kan det diskuteres om denne forskel i erfaring med programmering, kan skyldes forskellen i vurderingen. Ydermere kan man diskutere om prototypen henvender sig mere til de personer, der har et højere kendskab til programmering end tilfældet var med FP6. Målgruppen er sat til personer, der har en basal forståelse for programmering på et hobbyniveau. Her kan det diskuteres om FP6 er en del af denne målgruppe, når forsøgspersonen har vurderet egne evner til at være 20 på en skala, der går til 100.

Den gennemsnitlige vurdering af *Get started* guiden er til karakteren A. Ydermere er *How to* guiden vurderet ud fra den gennemsnitlige SUS score til karakteren A+ og ud fra medianen til karakteren A. Dette kan vidne om, at prototypen har opnået et tilfredsstillende niveau, således at hele målgruppen, kan benytte dokumentationen ud fra deres arbejdsmønstre. Dog er denne version af prototypen kun testede på fire personer og dette vil påvirke den eksterne validitet.

Der har ikke været optaget brugbar lyd i gennemførelserne for FP5, FP7 og FP8, hvilket kan have medført et tab af data, da der kun blev taget enkelte noter under debriefingen. Det ses dog ud fra deres kommentarer fra spørgeskemaet og forsøgspersonernes SUS score, at der generelt har været en positiv holdning til prototypen. Dette betyder dog ikke, at visse forbedringspunkter, som er blevet kommenteret, er gået tabt grundet den manglende lydoptagelse.

Forsøgspersonerne blev instrueret i kun, at forholde sig til en delopgave af gangen i SUS spørgeskemaet og under hver debriefing. Det kan dog ikke afvises, at forsøgspersonerne har oplevet en form for *carry over effect*, da deres bedømmelse under delopgave 2, kan være blevet påvirket af deres opleves af delopgave 1. Denne mistanke om *carry over*

effect, blev yderligere understøttet af FP7, der udtalte efter forsøget, at da han nu kendte systemet bedre, havde han nemmere ved at vurdere udsagnene fra SUS spørgeskemaet mere ekstremt. Dog vurdere FP5 prototypen til en SUS på 95 ved delopgave 1 og 100 ved delopgave 2.

Grundet det valgte forsøgsdesign, kan det diskuteres om der har været lav økologisk validitet i iterationerne generelt. Dette kan blandt andet være grundet den kunstige situation forsøgspersonerne blev sat i, hvor de blev observeret under forsøgset. Ydermere udtalte flere forsøgspersoner i debriefingen, at de vil have handlet anderledes, hvis det ikke var pga. forsøget. Her nævner FP7 f.eks. at han ville have benyttet videoerne i *How to* guiden, hvis det var under normale omstændigheder. Dog fik forsøgspersonerne mulighed for at benytte egne computere under forsøget, hvilket kan have været med til at normalisere situationen til en vis grad.

Der måles på tid for forsøgspersonerne, dog bliver forsøgspersonerne ikke instrueret i at skulle udføre deres projekt hurtigt. Ydermere bruger flere af forsøgspersonerne tid på, at prøve forskellige temperaturer af på Wappsto inden opgaven afsluttes og tiden har derfor generelt ikke været i fokus. Derfor kan man også argumentere for, at tiden ikke er et essentielt usability måling i denne kontekst, da den ikke skal sammenlignes med en anden version eller på anden bliver sat i kontekst.

5.5.4 Delkonklusion

Denne iteration af forsøg har givet det ønskede indblik i prototypens usability problemer, som vurderes at være af mindre betydning på dette stadie. Dette bliver vurderet grundet den gennemsnitlige SUS score, hvor opsætningsguiden fik A og *how to* guiden fik A+. Ydermere gav den modtaget feedback fra forsøgspersoner indsigt i optimeringsmuligheder der var af mindre prioritet.

I appendiks G findes en gennemgang af struktur og designvalg for den femte version af prototypen, hvilket danner grundlag for forskningsspørgsmål 4.

Grundet tidsøkonomiske begrænsninger vælges det ikke at videreudvikle mere på prototypen i dette projekt.

Diskussion 6

I det følgende kapitel vil design tilgangen for projektet blive diskuteres, herunder validitet og metodebrug.

I dette projekt er det valgt, at benytte sig af en brugercentreret tilgang, hvor der via brugerundersøgelser er blevet undersøgt, hvem målgruppen er og hvad de har af krav og forventninger til dokumentationen. Man kunne i stedet have valgt en *Activity Centred* tilgang. På denne måde kunne man have haft fokus på selve aktiviteten og hvilke steps det kræver for at løse denne. I projekts tilfælde vil aktiviteten f.eks. være at benytte biblioteket til at automatisere handlinger, hvoraf stepsene vil blandt andet være opsætning og eventuelt en *How to* guide. Norman [2013][s. 231] forklarer, at folk har en tendens til at være villige til at tillære sig evner, hvis de er essentielle for aktiviteten. Ud fra denne tilgang skulle der i designet således ikke tages forbehold for begyndere og erfarne, som der er gjort i dette projekt. Det kan dog diskuteres om denne tilgang er relevant for dette projekt, set ud fra resultaterne fra brugerundersøgelserne og den gennemgået teori. Her vil en *activity Centred* tilgang, kunne resultere i at informations niveauet i dokumentationen ikke vil blive niveauopdelt som nu, hvor den erfarne bruger kan skimte sig igennem og den mere uerfarne kan blive guidet via forskellige artefakter. Som beskrevet i afsnit 1.1 definerer Bhatti et al. [2021][s. 149 - 154] ydermere funktionel kvalitet til blandt andet, at der skal tages hensyn til brugerens læseniveau, samt hvor let brugeren kan navigere i dokumentationen. Da den *activity Centred* tilgang ikke fokuserer på brugernes mål og præferencer, men i stedet på aktiviteter, harmonere dette ikke med den funktionelle kvalitet på disse punkter. I afsnit 2.1, blev det ligeledes udledt at for at udarbejde effektiv dokumentation kræver det, at man opnår forståelse for brugerens behov og inddrager dem i designprocessen.

Igennem usability forsøgene beskrevet i kapitel 5, er der kun testet på folk der har vurderet dem selv til, at have en høj grad af ekspertise indenfor brug af computere generelt. Jf. tabel F.2 ses det, at majoriteten af forsøgspersonerne har vurderet deres egne programmeringsevner til at være omtrent middel. Det kan derfor diskuteres om det er lykkes, at teste prototypen på erfarne brugere, som ifølge den gennemgået teori i afsnit 1.1, vil have en anden tilgang til brug af biblioteket end uerfarne brugere. Dog kan dette modargumenteres med, at i prototypens indledende designproces, blev der benyttet flere karriereudviklere, samt domæneeksperter i udviklingen af prototypen som f.eks. en heuristisk evaluering og feedback fra udviklere. Til dette kan man diskutere om det havde været fordelagtigt, at varierer brugen af forsøgspersoner mere i designprocessen, således karriereudviklerne vil have været med i den afsluttede del af designprocessen.

Grundet projektet har taget udgangspunkt i WappstoIoT har det ikke været muligt at teste på eksterne domæneeksperter, da WappstoIoT på nuværende tidspunkt ikke har

nogle brugere. Det kan ydermere diskuteres om designeændringerne, som blev fortaget på baggrund af usability forsøgene i kapitel 5, har haft en signifikant indvirkning på det ønskede brug fra karriereudviklere. Dette vil man med fordel kunne teste ved at udføre en iteration, hvor det er netop denne del af målgruppen, der har til opgave at udføre delopgave 1 og 2, som beskrevet i kapitel 5.

Det er ikke blevet fundet relevant at benytte inferential statistik ved de benyttede kvantitative metoder, da der er blevet prioriteret iterationer over antallet af forsøgspersoner per iteration. Dette har haft den konsekvens, at der primært er truffet designændringer på baggrund af tendenser i stedet for signifikante forskelle. Derfor kan man argumentere for, at dette har påvirket den eksterne validitet under usability forsøgene og det udsendte spørgeskema i brugerundersøgelserne.

Projektets formål har været at udvikle et generisk design, der kan bruges til flere hjælper biblioteker, der har lignede karakter som WappstoIoT. Designet og strukturen er dog ikke blevet testet på andre lignede biblioteker. Dette kan derfor ligeledes have påvirket den eksterne validitet, da det ikke vides om designet og strukturen vil få lignede resultater som den udviklede prototype har fået.

Selvom den gennemsnitlige vurdering af prototypen, var af karakterene A og A+, viste resultaterne fra sidste iteration stadig, at der kan med fordel kan laves designmæssige ændringer. Ydermere kan der argumenteres for, at der er en lav ekstern validitet, da prototypen kun er testede i laboratorie lignede opsætninger, samt på en smal del af målgruppen. Dog har tidsøkonomiske begrænsninger gjort, at det ikke har været muligt at teste yderligere på prototypen.

Under hele designprocessen, er der kun blevet taget udgangspunkt i et Windows styresystem. Under usability forsøgene, udtrykte flere forsøgspersoner, at de primært brugte Linux og derfor var nogle kommandoer anderledes end de viste. Derfor kan det diskuteres om dokumentationen burde inkludere flere styresystemer eller fokusere på det styresystem, der primært bliver brugt af personer i målgruppen. Dog viste brugerundersøgelserne og den gennemgået teori, at mængden af information i dokumentationen skal holdes til kun det essentielle. Ydermere blev det observeret under usability forsøgene, at forsøgspersonerne der normalt bruger Linux styresystem var klar over disse forskelle og derfor selvstændigt ændrede kommandoerne uden større udfordringer. Ydermere forklarede flere af de personer som var vant til at bruge Linux, at det ikke var en større udfordring, at eksemplerne var vist på et Windows styresystem, da rækkefølgen og måden hvorpå man udfører opgaver er meget a lig.

Et af projekts formål var, at undersøge hvorledes man kunne integrerer dokumentation for flere programmeringssprog i samme design. Da WappstoIoT er udviklet til Python, Arduino og JavaScript blev der taget udgangspunkt i disse sprog. Til Arduino biblioteket skulle der dog bruges hardware i form af en ESP32 eller en WIO terminal. Grundet denne forskel blev Arduino eksemplet ikke indført i prototypen, da sammenligningen bibliotekerne imellem vil blive besværliggjort. Derfor kan man argumentere for, at prototypen på nuværende stadie ikke kan indholde biblioteker, som har væsentlige forskelle i udføreslen. Det er dog tiltænkt, at opsætning og måden hvorpå brugeren bliver guidet, skal have lignede karakter som med Python og JavaScript. Ved Arduino vil der dog skulle designes for fysiske handlinger og

opsætning af trejdeparts hardware.

Selvom Arduino delen af biblioteket aldrig blev integreret i prototypen, blev der stadig testet på Python og JavaScript. Ud fra tabel F.1 og de valgte sprog under iterationerne af usability test, har forsøgspersonerne valgt deres fortrukne sprog i opgaveløsningen. I introduktionen til forsøget blev der bevidst ikke fortalt noget om programmeringssprog, men der blev blot sat et scenarie op. At forsøgspersonerne selv har kunne vælge imellem programmeringssprogene kan vidne om en forståelse fra forsøgspersonerne om, at biblioteket kan virke til flere forskellige sprog. Forståelsen fra forsøgspersonen kan yderligere understøttes af, at flere forsøgspersoner først trykkede sig ind på de tre forskellige programmeringssprog for at skabe sig et overblik, inden der blev fortaget et valg.

Konklusion 7

Antallet af tredjeparts biblioteker har været stigende i de seneste år og det samme gælder for kompleksiteten af disse biblioteker. Derfor er det vigtigt, at dokumentationen af bibliotekerne er designet således at hele målgruppen kan benytte bibliotekerne på bedst mulig vis. Ud fra denne problemstilling, har projektet fulgt en designproces, hvor der først er dannet en grundviden omkring dokumentation og dens brugere. Denne grundviden har bl.a. inkluderet viden omkring de forskelle, der findes blandt personer med forskellige erfaringsniveauer og hvilke problemstillinger, der opstår grundet disse.

Ud fra denne viden blev der opstillet følgende problemformulering

> "*Hvordan kan brugerdokumentation udvikles, så det kan benyttes på tværs af programmeringssprog og af personer med forskellige programmeringsevner?*"

Med udgangspunkt i denne problemformulering, blev der udarbejdet fire forskningsspørgsmål:

- Forskningsspørgsmål 1: Hvordan anvender udviklere med forskellige programmeringsevner dokumentationen til tredjeparts biblioteker?
- Forskningsspørgsmål 2: Forskningsspørgsmål 2: Hvilke udfordringer opleves der i udviklingen af dokumentationen og i brugen af den?
- Forskningsspørgsmål 3: Hvilke ønsker har udviklere med forskellige programmeringsevner til indhold i dokumentationen for trejdeparts biblioteker?
- Forskningsspørgsmål 4: Hvilke designelementer, samt struktur skal dokumentationen af tredjeparts biblioteker have for, at tilgodese både erfarne og uerfarne brugers arbejdsmønstre?

For at kunne besvare forskningsspørgsmål 1, 2 og 3, blev der udført brugerundersøgelser i form af interviews og der blev udsendt et online spørgeskema. Disse undersøgelser resulterede i tre fokuspunkter, der havde til formål at konkretisere emner, der skulle være i særlig fokus i videreudviklingen af dokumentationen. Brugerundersøgelserne resulteret ligeledes i en liste med essentiel indhold til dokumentationen. Herfra blev projektet afgrænset til udelukkende at forholde sig til forsiden af dokumentationen, en opsætningsguide og en *how to* guide.

For at kunne besvare forskningsspørgsmål 4, blev der udviklet en prototype, hvor der blev fortaget forsøg og evalueringer af prototypen over fem iterationer. Disse iterationer mundede ud i en femte version af prototypen, som har til hensigt at illustrere et generisk design for dokumentation for trejdeparts hjælper biblioteker. Resultaterne fra usability

forsøgene viste, at opgaverne for opsætningsguiden og *How to* guiden, blev vurderet til en gennemsnitlig SUS score på hhv. 83,75 og 85,63, hvilket ifølge Jeff Sauro [2018] vil give karakterne A og A+.

Projekt har derved undersøgt hvordan man kan udvikle og designe dokumentation, som kan tilpasses til personer med forskellige programmeringsevner. Ydermere har projektet undersøgt, hvordan man kan integrerer flere programmeringssprog i dokumentationen af tredjeparts biblioteker.

Dog skal dette projekt ikke ses som et færdigt resultat, da der stadig mangler at blive undersøgt, hvorledes den resterende del af dokumentationen bør designes. Ligeledes mangler projektet at undersøge, hvorledes prototypens struktur og design passer til andre biblioteker end WappstoIoT, samt at teste prototypen på en større del af målgruppen.

Litteratur

Aurora Harley (2018). *Heuristic Evaluation.* `https://www.nngroup.com/articles/ux-expert-reviews/`.

Ao, W. and Jamir, B. (2020). Application of garret ranking technique in studying the problems of bamboo cultivation: A case study of mokokchung district, nagaland. *Indian Journal of Hill Farming.*

Bhatti, J., Corleissen, Z., Lambourne, J., Nunez, D., and Waterhouse, H. (2021). *Docs for Developers.* Number ISBN 978-1-4842-7217-6 in Paperback. Apress.

Braun, V. and Clarke, V. (2006). Using thematic analysis in psychology. *Qualitative Research in Psychology*, 3:77–101.

Brinkmann, S. and Tanggaard, L., editors (2015). *Kvalitative metoder: En grundbog*, volume 1. Hans Reitzels Forlag, Danmark, 2 edition.

Edward de Bono Group (2017). *Six Thinking Hats.* `http://www.debonogroup.com/services/core-programs/six-thinking-hats/`.

FITTS, P. and POSNER, M. (1967). *Human performance.* Brooks/Cole Pub. Co., Belmont, Calif.

Garrett, H. E. (1940). *Statistics in psychology and education.* New York, London Longmans, hard copy edition.

Geiger, R. S., Varoquaux, N., and Mazel-Cabasse, C. (2018). The types, roles, and practices of documentation in data analytics open source software libraries. comput supported coop. *Berkeley Institute for Data Science.*

Gillham, B. (2008). *Developing a Questionnaire.* Number ISBN 9780826496317 in Paperback. Bloomsbury Publishing Plc.

Guest, G., Bunce, A., and Johnson, L. (2006). How many interviews are enough? *Field Methods - FIELD METHOD*, 18:59–82.

Interaction Design Foundation (2017). *Heuristic Evaluation.* `https://www.interaction-design.org/literature/topics/heuristic-evaluation`.

Ismanto, H., Suharto, Azhari, and Arsyad, L. (2018). Ranking method in group decision support to determine the regional prioritized areas and leading sectors using garrett score. *International Journal of Advanced Computer Science and Applications.*

Jakob Nielsen (1994). *10 Usability Heuristics for User Interface Design.* `https://www.nngroup.com/articles/ten-usability-heuristics/`.

Jakob Nielsen (1997). *The Use and Misuse of Focus Groups.* https://www.nngroup.com/articles/focus-groups/.

Jakob Nielsen (2000). *Why You Only Need to Test with 5 Users.* https://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users/.

Jason Warner (2018). *Thank you for 100 million repositories.* https://github.blog/2018-11-08-100m-repos/ Læst: 10-02-2022.

Jeff Sauro (2018). *5 Ways to Interpret a SUS Score.* https://measuringu.com/interpret-sus-score/.

Kate Kaplan (2017). *Facilitating an Effective Design Studio Workshop.* https://www.nngroup.com/articles/facilitating-design-studio-workshop/.

Kula German, D.M. Ouni A., e. a. (2018). Do developers update their library dependencies? *JEmpir Software Eng 23, 384–417.*

Larios Vargas, E., Aniche, M., Treude, C., Bruntink, M., and Gousios, G. (2020). *Selecting Third-Party Libraries: The Practitioners' Perspective*, page 245–256. Association for Computing Machinery, New York, NY, USA.

Nielsen, J. (1993). *Usability Engineering.* Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Nikola Kolev (2017). *The pros and cons of using a framework.* https://jaxenter.com/pros-cons-using-framework-135901.html Læst: 07-02-2022.

Norman, D. (2013). *The Design of everyday thing - Revised and Exanded edition.* Number ISBN: 978-0-465-05065-9 in Paperback. Basic Books.

Parnas, D. L. (2003). *The Future of Software Engineering.* Number ISBN 978-3-642-15186-6 in Paperback. Springer.

Preece, J., Rogers, Y., and Sharp, H. (2015). *Interaction Design: Beyond Human-Computer Interaction.* Wiley, 4 edition.

Proctor, R. and Zandt, T. (2008). *Human factors in simple and complex systems.* CRC Press, Boca Raton, FL, elektronisk udgave. -2nd ed. edition.

Rasmussen, J. (1986). *Information Processing and Human-Machine Interaction: An Approach to Cognitive Engineering.* Elsevier Science Inc., USA.

Rikke Friis Dam (2021). *Test Your Prototypes: How to Gather Feedback and Maximise Learning.* https://www.interaction-design.org/literature/article/test-your-prototypes-how-to-gather-feedback-and-maximise-learning.

Santhanam, R. and Wiedenbeck, S. (1993). Neither novice nor expert: The discretionary user of software. *Int. J. Man Mach. Stud.*, 38:201–229.

Satish, C. J. and Anand, M. (2016). Software documentation management issues and practices: a survey. *Indian Journal of Science and Technology*, 9(20).

Seluxit (2017). *Wappsto Unified Data Model.* `https://wappsto.seluxit.com/documentation/wappsto-unified-data-model/`.

Seluxit A/S (2021). *Seluxit Internet of Things.* `https://www.seluxit.com/about-seluxit/customers/customer-and-projects/`.

Shneiderman, B. (2004). *Designing the User Interface: Strategies for Effective Human-Computer Interaction.* Addison-Wesley Longman Publishing Co., Inc., USA, 4rd edition.

Thoriq Firdaus (2021). *The Importance of Documentation for Web Developers.* `hongkiat.com/blog/why-documentation-essential/` Læst: 12-02-2022.

Toby Osbourn (2019). *The Long-Term Impact of Great Code Documentation (with Examples.* `https://textexpander.com/blog/code-documentation` Læst: 13-02-2022.

Interviewguide A

Formålet med dette kapitel er at beskrive og argumentere for, hvordan interviewene med udviklerne fra Seluxit er blevet afholdt, samt hvilke spørgsmål der vil danne grundlag for interviewet.

Interviewguiden vil have forskningsspørgsmålene beskrevet i kapitel 2.5 som udgangspunkt for interviewet. Ud fra disse forskningsspørgsmål er der blevet udarbejdet interviewspørgsmål.

Grunden til at interviewguiden opdeler forskningsspørgsmål og interviewspørgsmål er, at ifølge Brinkmann and Tanggaard [2015][s. 40] dækker de to typer spørgsmål over forskellige hensigter. Forskningsspørgsmål er generelt på et mere abstrakt plan, som søger forklaringer på processer og sammenhænge og de fungerer derfor sjældent som spørgsmål til respondenten.
Et interviewspørgsmål bør derimod være mere mundrette, ligefremme og søge beskrivelser på konkrete processer og sammenhænge.

Selvom der er udarbejdet både forskningsspørgsmål og interviewspørgsmål, gør dette ikke at der ikke kan afviges fra dem under interviewsene. Dette er grundet, at man ifølge Brinkmann and Tanggaard [2015][s. 38] skal være kreativ som interviewer under interviewet, dog inden for den ramme man har etableret igennem forberedelse og udarbejdelse af interviewguide

A.1 Interviewguide

Følgende afsnit vil gennemgå den benyttede interviewguide.

Introduktion til interviewet:

Introduktionen forventes ikke at blive sagt ordret, men alle emner i introduktionen vil blive berørt.

Introduktionen lyder som følger:
Mange tak for din deltagelse i interviewet. Interviewet omhandler dine erfaringer og holdninger til brug og udvikling af softwaredokumentation. Interviewet vil være delt op i to dele, hvor vi først vil snakke om dig som udvikler af et bibliotek og hvilke erfaringer du har der og dernæst vil samtalen gå på dig som bruger af tredjepartsbilioteker og de erfaringer du må have med disse.

Vores udgangspunkt i hele interviewet vil være dokumentation til slutbrugeren. Interviewet vil blive lydoptaget, hvilket du er blevet gjort bekendt med via samtykkeerklæringen.

Medmindre du ønsker andet, vil dit navn og arbejdstitel fremgå i rapporten. Rapporten og lydoptagelser vil blive opbevaret fortroligt. Jeg forventer, at interviewet maksimalt vil tage en time. Nogle spørgsmål?

Interviewet starter herefter op med en række åbningsspørgsmål, der har til formål at udlede informationer om erfaring indenfor programmering, samt skabe en fælles forståelse for hvad softwarebiblioteker og dokumentation af dette er.

Basale spørgsmål

Åbningsspørgsmålene skal ses som indledende spørgsmål til de næste to dele af interviewet. Her vil del 1 være spørgsmål, der er til formål at få indsigt i udvikleren af et tredje softwarebibliotek, hvor anden del vil være fokuseret på slutbrugeren.

Det vurderes at de interviewede kan sætte sig i begge roller, da de til dagligt både arbejder med at udvikle projekter selv og derved benytter tredjepart biblioteker, men også selv har stået for udviklingen og dokumentationen af et tredjepart bibliotek.

Følgende åbningsspørgsmål er blevet udarbejdet:

- Vil du fortælle lidt om dig selv og dine erfaringer med programmering?
- Hvad arbejder du til dagligt med?
- Hvilke programmeringssprog skriver du primært i?
- Hvordan vil du forklare, hvad tredjepart softwarebiblioteker er?
- Kan du prøve at give mig et eksempel på et tredjepart softwarebibliotek du har brugt for nyligt?
- Hvad var formålet med dette?
- Vil du fortælle lidt om hvilke tredjepart biblioteker du tidligere har været med til at udvikle?
- Hvad har været deres formål?
- Vil du prøve at forklare lidt om, hvad dokumentation af et bibliotek er?
- Hvilke former for dokumentation har du arbejdet med i udviklingen af disse biblioteker?
- Hvilke målgrupper har disse biblioteker haft?

Del 1: Spørgsmål til udvikleren

Den første del efter åbningsspørgsmålene har til formål, at få indsigt i respondentens erfaringer med udviklingsprocessen af et tredjeparts bibliotek, samt dokumentationen af dette. Følgende spørgsmål er blevet udarbejdet til dette formål:

Bibliotek

- Prøv at beskriv hvilke scenarier der kunne gøre, at du som udvikler vil lave et trejdepart bibliotek?
- Hvilke parametre vægter du højt, når du udvikler et tredjepart bibliotek?
- Hvad har du af forventning til fagligheden ved brugeren til dit bibliotek?
- Følger du nogle standarder i udviklingen af

Dokumentation

- Når du høre ordet dokumentation som udvikler hvad tænker du så?
- Hvad vil du mene at formålet med dokumentation generelt er?
- Hvad mener du formålet med dokumentation til slutbrugeren er?
- Hvor stor en prioritet i dit daglige arbejde er dokumentation af din kode for dig?
- Hvad med for din arbejdsplads?
- Hvornår i udviklingsprocessen begynder du at dokumentere din kode?
- Hvilke former for dokumentation, er det ved de forskellige stadier?
- Prøv at beskriv hvem du har skrevet dokumentationen til ved sidste færdiggjort projekts?
- Hvor lang tid har du brugt på dokumentationen til brugeren?
- Hvilke dele af dokumentationen til slutbrugeren plejer du at lægge særlig vægt på?
- I hvilket format har du tidligere skrevet din dokumentation i?
- Hvor lang tid plejer du at bruge på at dokumentere din kode?
- Hvor lang tid planlægger udviklingsholdet på det?
- Følger din dokumentation nogle standarder, når det gælder dokumentation til slutbrugeren?
- Hvor plejer du at placere dokumentationen? (i koden, ekstern hjemmeside)
- Hvor har du fundet inspiration til udførelsen af dokumentationen?
- Hvor høj en prioritet har du oplevet at dokumentation til slutbrugeren har?
- Bruger du nogle online værktøjer til udformingen af dokumentationen?

Del 2: Spørgsmål til brugeren
Anden del af interviewet har til formål, at få indsigt respondentens erfaringer med tredjepart biblioteker og brugen af deres dokumentation.
Her vil spørgsmålene først være til general brug, hvorefter det ønskes undersøgt hvilke erfaringer og holdninger, der er til vante biblioteker, samt når der er blevet brugt nye biblioteker.

Følgende spørgsmål er blevet udarbejdet til dette formål: **Bibliotek**

- Hvor ofte bruger du trejdepart biblioteker i dine projekter?
- Med hvilket formål vil du bruge et trejdepart bibliotek i et projekt du er igang med?
- Hvilke udfordringer oplever du som oftes ved brugen af trejdepart biblioteker?
- Hvordan beslutter du dig for om du vil bruge et bibliotek eller ikke?

Del 2: Spørgsmål til brugeren

Dokumentation

- Med hvilket formål bruger du dokumentation generel?
- Hvad forventer du som slutbruger af indholdet for dokumentationen?
- Hvor vil du forvente at finde dokumentationen (f.eks. i koden, ekstern hjemmeside)?
- Hvad vil du mene gør en kode veldokumenteret?
- Hvis du skulle nævne de bedste veldokumenteret biblioteker du har brugt, hvilke er det så?
- Hvornår bruger du dokumentation som opslagsværk?
- Hvad forventer du af indhold til dokumentationen som opslagsværk?
- Hvornår bruger du dokumentation til at guide dig igennem en proces?
- Hvad forventer du af indhold til dokumentationen som en guide?

- Hvordan ser du dokumentation til slutbrugeren i tredjeparts biblioteker ser ud nu?
- Hvordan vil du gerne have den til at se ud?

Bemærkninger til intervieweren

- Husk at tage papir og blyant med, så den interviewede har mulighed for at tegne og forklare.
- Husk samtykkeerklæringer og at disse skal være underskrevet inden optagelsesstart
- Hold en afslappet stemning i interviewet og fokusere på at være aktiv lytter.

Udvikling af spørgeskemaet B

Dette appendiks har til formål gennemgå fremgangsmåden i udsendelse af spørgeskemaet, samt at præsentere de samlede resultater fra spørgeskemaet.

Spørgeskemaet er udarbejdet i SurveyXact[1] og er blevet lagt op i otte forskellige Facebook grupper.

Udvælgelsen af Facebook grupperne er sket på baggrund af, hvilken målgruppe Facebook gruppen måtte antages at tiltrække. F.eks. er spørgeskemaet udsendt i grupperne "Unge Programmører", som har 9.000 medlemmer og "Arduino for Beginners", som har 16.900 medlemmer[2]. Disse grupper er valgt, da det forventes at der er både udviklere med erfaring, men også udviklere på begynderniveau. Vurderingen af grupperne er sket på baggrund af en subjekt vurdering, hvor tilgangen har været at nå ud til flest mulige indenfor målgruppen.

Der er udarbejdet to versioner af et Facebook opslag, den ene på dansk og det andet på engelsk. Dette er valgt, da en gruppe som "Unge programmører"må forventes primært at have danske medlemmer, hvorimod en gruppe som "Arduino for Beginners", må forventes at have medlemmer fra adskillige lande.

De to forskellige versioner af Facebook opslaget var som følger:

Engelsk version
Hi developers at all levels!

I need your expertise!
I am currently working on my master's thesis in Engineering Psychology at Aalborg University.

The thesis purpose is to investigate how to document third-party libraries, so the documentation can be used in a simple and intuitive way for developers at all levels and still contain all relevant content.
For this purpose, I have created this questionnaire, which will take approximately 5 minutes to complete.
Link to survey: https://www.survey-xact.dk/LinkCollector?key=7LVSRREXUNCJ Please feel free to share this survey.
Thank you so much for your help, your help is highly appreciated.

Dansk version

[1] `https://www.survey-xact.dk/`
[2] læst d. 25/3-2022

Hej allesammen
Jeg arbejder i øjeblikket på mit speciale i Engineering Psychology på Aalborg Universitet.

Formålet med specialet er at undersøge, hvordan man kan dokumentere tredjepartsbiblioteker, så dokumentationen kan bruges på en enkel og intuitiv måde for udviklere på alle niveauer og stadig indeholde alt relevant indhold.

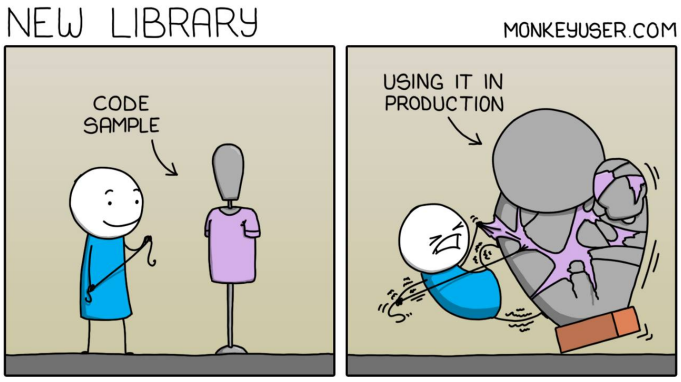
Til dette formål har jeg lavet et spørgeskema, som vil tage imellem 5-10 minutter at udfylde og alle er velkommen til at svare på det uagtet programmeringsniveau. Jeg håber i har lyst til at hjælpe mig!

I kan tilgå spørgeskemaet her: https://www.survey-xact.dk/LinkCollector?key=7LVSRREXUNCJ

På forhånd mange tak for din hjælp, den er meget værdsat.
Del gerne denne undersøgelse.

Jeg håber mit opslag er ok ellers må admin slette det.

Begge opslag blev efterfulgt af et meme med det formål at kunne tiltrække folks opmærksomhed til opslaget. Memet var som følger:



Selve spørgeskemaet er kun udarbejdet i en engelsk version, da det forventes at man som udvikler primært arbejder med programmering på engelsk i den vestlige del af verden. Denne forventning kommer ud fra resultaterne fra interviewene beskrevet i afsnit 3.1.

En tekstudgave af spørgeskemaet kan findes i bilag A i filmappen "Spørgeskema". Ellers vil overskifterne for resultaterne i afsnit B.1 være de givet spørgsmål.

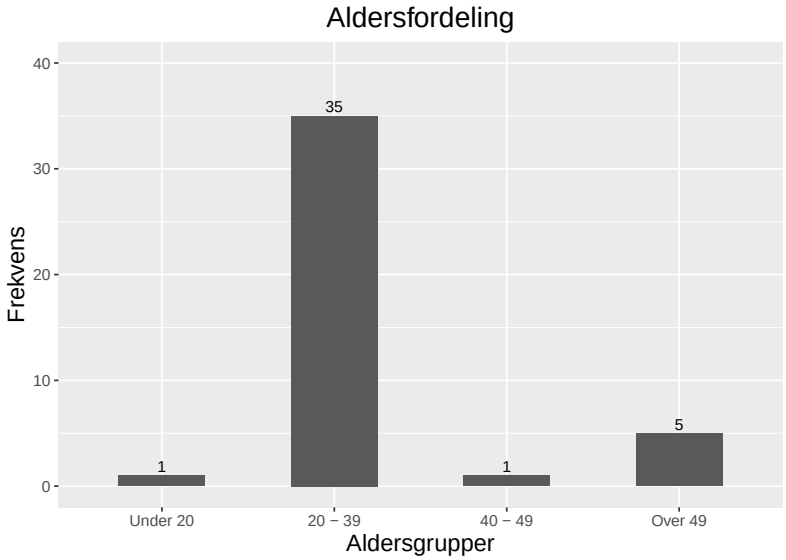
B.1 Resultater - Samlet

Følgende afsnit har til formål at undersøge de samlede resultater fra spørgeskemaet.

I dette afsnit vil det aktuelle spørgsmål fra spørgeskemaet være overskriften, hvorefter resultatet vil blive beskrevet og/eller illustreret. Fælles for alle figurer er, at deres kategorier og emner er oversat fra engelsk til dansk ift. selve spørgeskemaet.

Rådata kan findes i Appendiks B under filen "Spørgeskema". Spørgeskemaet er i alt besvaret af 41 respondenter hvoraf otte kun delvis gennemførte spørgeskemaet.

What is your age?



Figur B.1. Figuren viser aldersfordelingen af respondenterne. Henad x-aksen er aldersgrupperne og opad y-aksen er antallet af respondenter der ligger inden for aldersgruppen.

Som det ses på figur B.1, var der 35 respondenter der var i alderen 20 - 39 år, hvilket gør det til den den primære aldersgruppe. Den sekundære aldersgruppe var "Over 49"med fem respondenter, hvor de to sidste aldersgrupper havde en respondent hver.

What gender do you identify as?

- Mand: 35
- Kvinde: 2
- Andet: 4

Som det ses på ovenstående tabel, var 35 af respondenterne mænd, to var kvinder og fire ønskede enten ikke at svare på spørgsmålet eller fandt det ikke relevant at svare på.

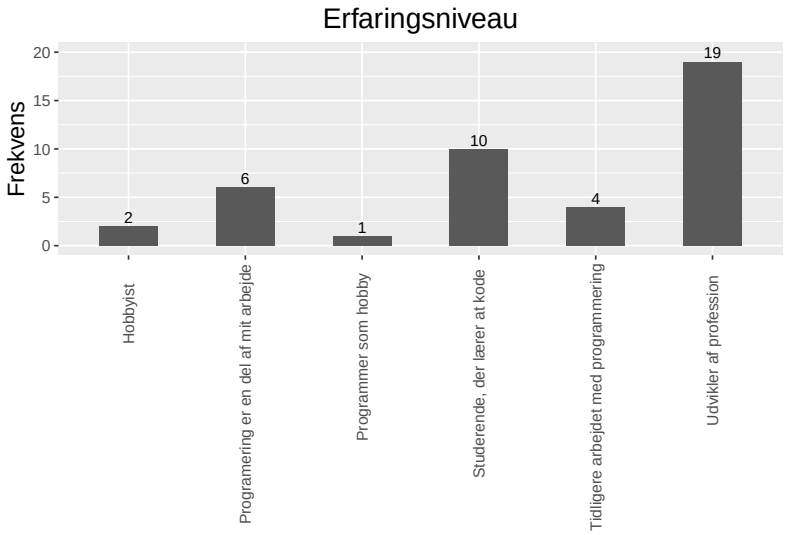
Where do you live?

- Danmark: 35
- Frankrig: 2
- Holland: 1
- USA: 4
- Ikke angivet: 2

Ovenstående tabel viser at hovedparten af respondenterne er bosiddende i Danmark, men der var også respondenter fra Frankrig, Holland og USA. To respondenter har ikke ønsket at angive, hvor de boede.

Which of the following options best describes you?

Dette spørgsmål har til hensigt, at indsamle informationer om hvor hyppigt respondenterne benytter programmering i deres dagligdag og hvad deres erfaringsniveau er.

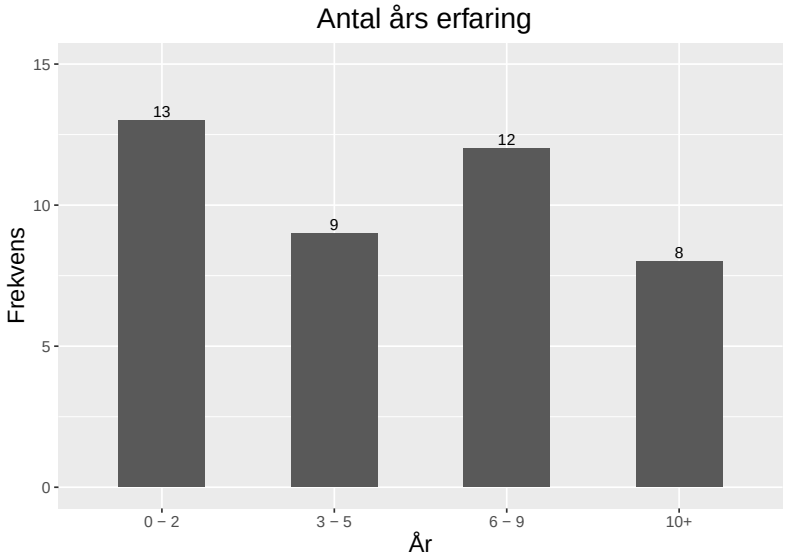


Figur B.2. Figuren viser, hvordan respondenterne selv ser deres erfaringsniveau. Henad x-aksen er kategorierne for erfaring. Denne akse er blevet oversat fra engelsk til dansk ift. selve spørgeskemaet. Opad y-aksen ses antallet af respondenter der hører under kategorien.

På figur B.2 ses det, at med 19 respondenter, der har programmering som profession, er dette den primære gruppe af respondenter. Den anden højeste gruppe af respondenter er studerende, der er i gang med at lære at kode.

How many years of experience do you have?

Dette spørgsmål har til formål at udlede respondenternes erfaringsniveau, hvoraf de har kunne vælge imellem fire forskellige intervaller.



Figur B.3. Figuren viser, hvor mange års erfaring respondenterne har med programmering. Henad x-aksen er års intervallerne og opad y-aksen er frekvensen.

På figur B.3 ses det, at der er flest respondenter repræsenteret i gruppen, der har nul til

to års erfaring. Dernæst kommer gruppen med seks til ni års erfaring.

Do you in general use third-party libraries in your projects?

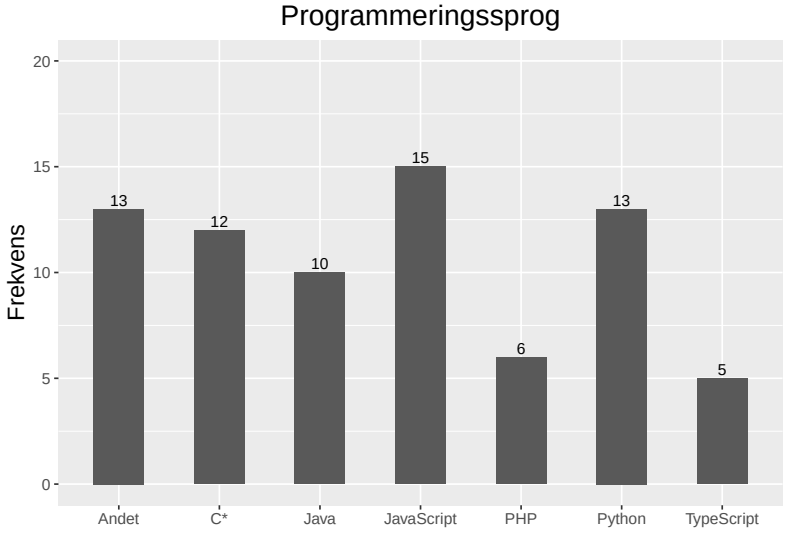
For at kunne undersøge om respondenterne generelt bruger trejdeparts biblioteker er dette spørgsmål blevet udarbejdet.

- Ja: 38
- Nej: 4

Ud fra ovenstående tabel, ses det, at der er 38 respondenter der bruger trejdepartsbiblioteker generelt i deres projekter og der er 4 der ikke gør.

Which programming languages do you primarily work in?

Da projektet ønsker at udforme dokumentation på tværs af programmeringssprog, er der blevet spurgt ind til respondenternes primære programmeringssprog. Flere respondenterne har svaret, at de har flere primære programmeringssprog, og derfor indgår de i flere af søjlerne på nedenstående figur.

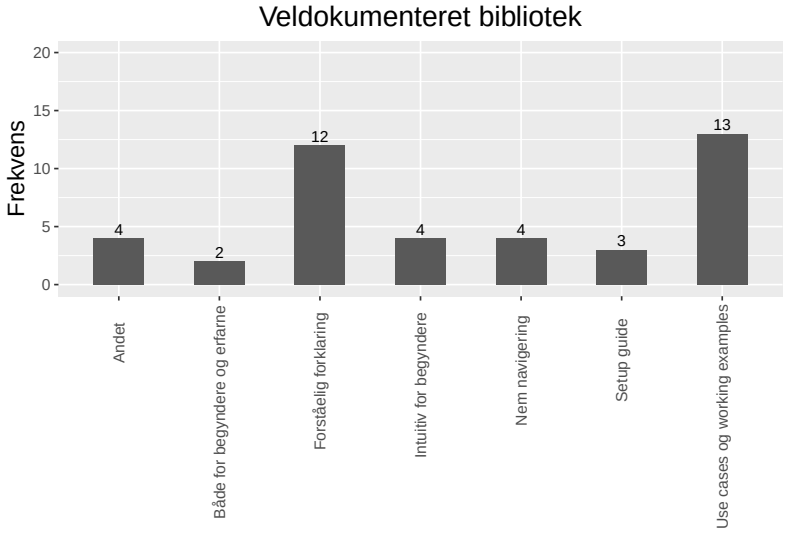


Figur B.4. Figuren viser, hvilke programmeringssprog respondenterne primært arbejder med. Der er flere respondenter, der har svaret med flere sprog. Henad x-aksen er programmeringssprog og opad y-aksen er frekvensen.

Som det ses på figur B.4 er det tre mest benyttede sprog af respondenterne C, Python og JavaScript. På figur B.4 dækker "Andet" over programmeringssprog som f.eks. MatLab, Ruby, node og Unity. For at se den komplette liste af programmeringssprog, der hører under "Andet", se bilag A i mappen “Spørgeskema”. På selvsamme figur dækker søjlen med C over typerne C, C Sharp, C++ og C#.

In a few sentences please describe what you consider to be a well-documented library

Dette spørgsmål har til formål, at undersøge hvad respondenterne definere som et veldokumenteret bibliotek. Respondenterne skulle i dette spørgsmål afgive deres svar ved at skrive i et tekstfelt. Derfor er svarene blevet samlet i emner, som er blevet udarbejdet via en personlig vurdering af hver udsagn.



Figur B.5. Figuren viser, hvor et overblik over hvad respondenterne mener er et veldokumenteret bibliotek. Henad x-aksen er emnet og opad y-aksen er frekvensen.

På figur B.5 ses en opsummering af hvilke emner respondenterne fandt vigtige ift. et veldokumenteret bibliotek. Under emnet "Andet"høre følgende under:

- Factory pattern libs
- and which is up to date.
- A library with an online presence, as well as online documentation. Preferably user-contributed. In-editor documentation such as typescript type-definitions are the most important. Finally, an active community.
- If it has a read the docs page, a well formed github page. Code documentation in the form of comments an ide is able to show.

Da respondenterne skulle svare via fritekst var der op til flere respondenter, der gav flere eksempler, hvilket har medført at de er repræsenteret i flere af emnerne i barplottet.

Som det ses på figur B.5 har respondenterne primært nævnt use cases og *working examples* som værende parametre for et veldokumenteret bibliotek. Dernæst kommer en "Forståelig forklaring", som både indeholder beskrivelse af formål, samt brug af funktioner og variabler.

In general where would you expect to find the documentation for a third party library?

Formålet med dette spørgsmål er at undersøge, hvor målgruppen vil forvente at finde dokumentationen. Respondenterne har haft muligheden for at vælge GitHub eller en

dedikeret hjemmeside eller skrive i feltet ved *Other*.

- GitHub (or similar platforms): 19
- A dedicated website: 11
- Other: 8

I svarmuligheden "Other"skrev respondenterne følgende:

- Begge dele kan være lige gode
- Both
- Depending on size github markdown or website
- Any website, as long as it's easily readable and uses table of contents.
- All of the above. If it's not well documented, only the most hard-core developers can see through it and thus 99% wouldn't use it
- Their should be a programing reference with the language and that reference should have illustrative examples of code usage.
- I have no expectations. I just Google it.
- Tutorial sites (not dedicated to specific library)

Ud fra resultaterne på de to ovenstående tabel, ses det at de fleste respondenter vil forvente at finde dokumentation på GitHub eller lignede platforme og dernæst kommer en dedikeret hjemmeside.

Where would you prefer to find the documentation?

Formålet med dette spørgsmål er at undersøge, hvor målgruppen vil fortrække at finde dokumentationen. Respondenterne har haft muligheden for at vælge GitHub eller en dedikeret hjemmeside eller skrive i feltet ved *Other*.

- GitHub (eller lignede platforme): 13
- En dedikeret hjemmeside: 17
- Andet: 8

I svarmuligheden "Andet"skrev respondenterne følgende:

- Begge dele kan være lige gode
- Doesn't matter much to me.
- From the git repository; it should either contain it as markdown or efternavn link to a website
- Read previous answer
- All of the above. If there is no good documentation in not use the code
- Their should be a programing reference with the language and that reference should have illustrative examples of code usage.
- A dedicated website if no code examples, otherwise github or similar with examples and docs written in a readme file.
- Tutorial sites

Ud fra resultaterne der ses på de to ovenstående tabeller, at de fleste respondenter vil fortrække at finde dokumentation på en dedikeret hjemmeside og dernæst kommer GitHub eller lignede platforme.

Please rank what you in general find most important for documentation for a library

Formålet med dette spørgsmål er at få indsigt i hvordan respondenterne prioriterer forskellige typer indhold i dokumentationen. Respondenten har fået til opgave at rangere fem forskellige emner.

For at finde ud af, hvordan respondenterne har rangeret denne opgave, benyttes der af *Garret Ranking Technique* [Garrett, 1940][s. 327 - 335].
Denne teknik har til formål at udregne rangeringen af emner på en systematisk måde. Teknikken er valgt, da den er blevet brugt i andre undersøgelser med rangeringsdata, som f.eks. Ao and Jamir [2020] og Ismanto et al. [2018].

Måden hvorpå *Garret Ranking Technique* udregnes, er ved først at tælle i datasættet, det antal gange som hvert emne er blevet rangeret i en bestemt orden.

Labels	Rank 1	Rank 2	Rank 3	Rank 4	Rank 5
Nem setup	12	9	6	8	2
Worked-examples/Kode snippets	12	12	8	4	1
Dybdegående dokumentation	2	6	11	8	10
Komplette eksempler	7	5	9	11	5
Dokumentation i kildekode	4	5	3	6	19

Tabel B.1. Tabellen viser det antal gange hvert emne er blevet rangeret i orden et til fem. Emner der er rangeret ved "Rank 1" er rangeret højest, hvorimod emner der er rangeret ved Rank 5 er rangeret lavest

Denne optælling ses på tabel B.1, hvor f.eks. “Nem setup” og “Worked-examples/Kode snippets” er blevet rangeret højest (Rank 1) af syv respondenter hver og “Komplette eksempler” er rangeret tredjehøjst fem gange.
Dernæst findes *procent position* værdien ved brug af følgende formel:

$$\frac{100(R_ij - 0,5)}{N_j} \tag{B.1}$$

Hvor:

- R_ij: Rangering givet for den ith variabel af jth respondenter
- N_j: Antal variabler rangeret af jth respondenter

Denne værdi skal findes for det antal rangeringsmuligheder som respondenten har haft. Det vil sige at respondenten har i dette spørgsmål skulle rangere fem forskellige emner i dette spørgsmål og derfor skal der findes fem procent positions værdier.
Efter have have fundet procent position værdien for hver rangering, kaldet Factor nummer i tabel B.2, skal man via tabel opslag finde *Garrett værdien* [Garrett, 1940][s. 329].

Faktor Nummer	Procent position værdi	Garrett værdi
1	10	75
2	30	60
3	50	50
4	70	40
5	90	24

Tabel B.2. Tabellen viser de udregnet "Procent position" værdier, samt Garrtt værdien som er fundet via tabelopslag.

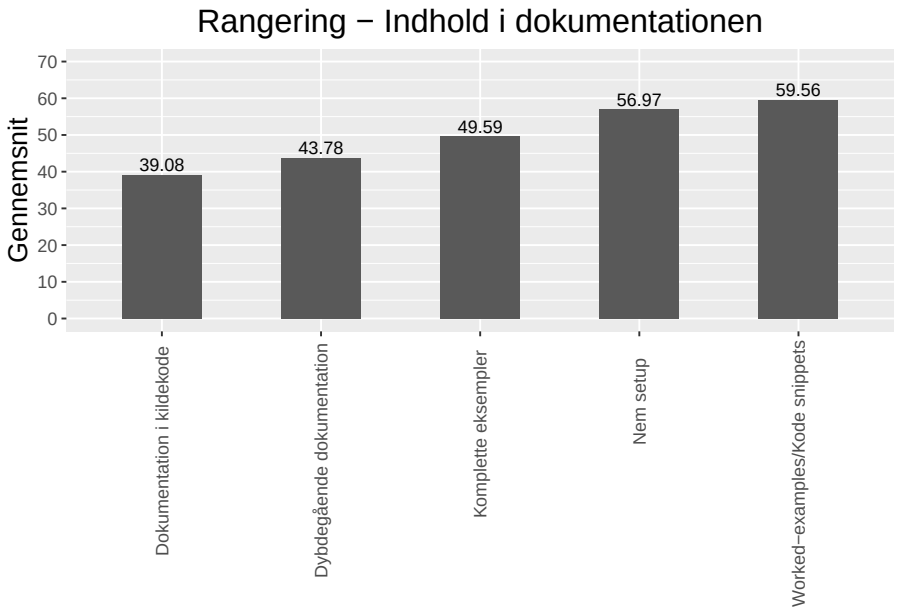
Garrett værdien bruges nu til at gange på hver frekvens i søjle to til seks i tabel B.1. Med dette menes der, at f.eks. bliver alle tal i søjle to på tabel B.1 ganget med Garrett værdien for faktor nummer 1, hvilket er 75 (se tabel B.2) og alle tal i søjle tre på samme tabel bliver ganget med 60.

Labels	Rank 1	Rank 2	Rank 3	Rank 4	Rank 5	Total	Garrett Gnm.	Rank
Nem setup	900	540	300	320	48	2108	56,97297	2
Worked-examples/Kode snippets	900	720	400	160	24	2204	59,56757	1
Dybdegående dokumentation	150	360	550	320	240	1620	43,78378	4
Komplette eksempler	525	300	450	440	120	1835	49,59459	3
Dokumentation i kildekode	300	300	150	240	456	1446	39,08108	5

Tabel B.3. Tabellen viser de nødvendige værdier for at kunne udregne Garrett gennemsnittet. Fra søjle to til seks ses tallet, hvor frekvensen fra tabel B.1 er ganget på den tilhørende Garrett værdi, som ses i tabel B.2. Søjlen "Total" viser den adderet sum og "Garrett Gnm." viser den udregnet gennemsnit. Søjlen yderst til højre viser hvordan de forskellige emner er rangeret.

Resultatet af dette ses i søjle to til seks på figur B.3. Herefter summeres tallene i rækken op, hvilket kan ses på tabel B.3 under “Total”. Dernæst divideres tallet med antallet af respondenter, hvilket i dette spørgsmål har været 34. Dette gøres for at finde den gennemsnitlige Garrett værdi for hvert emne, som er et udtryk for emnets rangering. Som det ses på figur B.3 ved søjle ni, kan man nu rangerer sine emner ud fra respondenternes svar.

På denne måde er alle rangeringsopgaver i dette spørgeskema blevet udregnet. Udregningerne kan ses i bilag A i mappen "Spørgeskema".

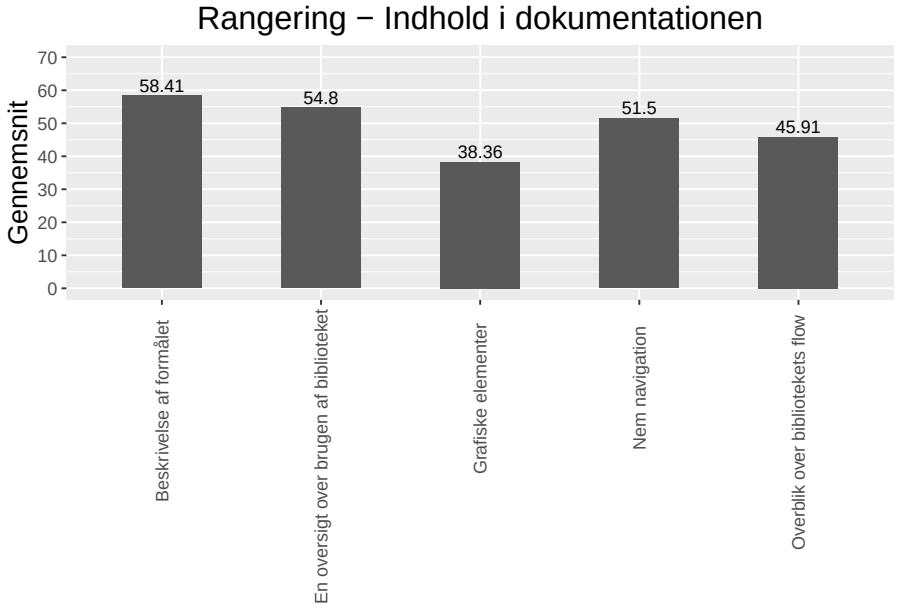


Figur B.6. Figuren viser respondentenes rangering ift. faktorer der influerer i valget af biblioteket. Henad x-aksen står emnet og opad y-aksen er gennemsnittet.

Figur B.6 viser rangeringen af spørgsmålet fra alle respondenterne. Her ses det, at det højst rangeret emne er *Worked-examples/kode snippets*, hvorefter *Nem setup* kommer. Ydermere ses det på tabel B.6, at det laveste rangeret emne er *Dokumentation i kildekode*

Please rank what you in general find most important for documentation for a library

Formålet med dette spørgsmål er, at få indsigt i hvordan respondenterne prioriterer forskellige typer af indhold i dokumentationen. Her har respondenterne ligesom ved sidste spørgsmål haft til opgave at rangere fem udsagn og udregningerne er fortaget på samme måde, som gennemgået ved ligning B.1. De eksakte udregninger kan ses i bilag B under filnavnet "Rangeringsopgaver".



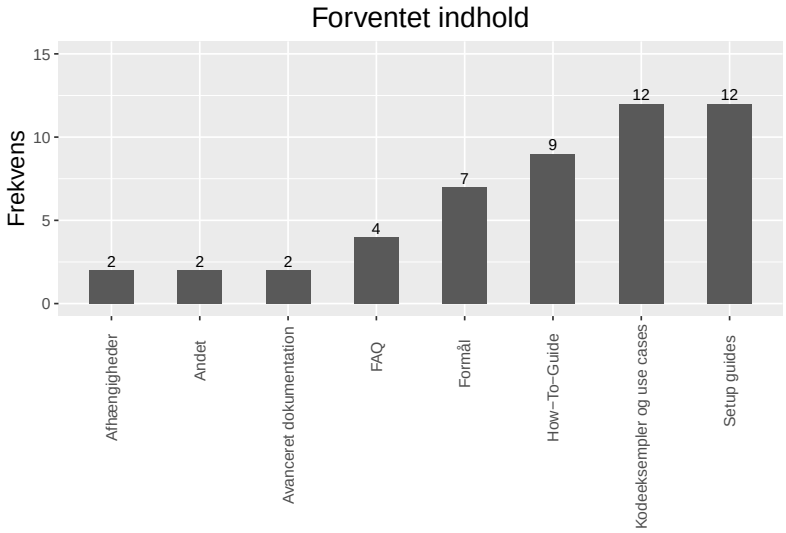
Figur B.7. Figuren viser respondentenes rangering ift. fortrukket indhold. Henad x-aksen ses emnet og opad y-aksen er gennemsnittet

Labels	Rank 1	Rank 2	Rank 3	Rank 4	Rank 5	Total	Garrett Gnm.	Rank
Overblik over bibliotekets flow	225	420	400	440	168	1653	45,91667	4
En oversigt over brugen af biblioteket	525	720	400	280	48	1973	54,80556	2
Beskrivelse af formålet	975	360	600	120	48	2103	58,41667	1
Grafiske elementer	375	60	250	240	456	1381	38,36111	5
Nem navigation	600	600	150	360	144	1854	51,5	3

Tabel B.4. Tabellen viser de nødvendige værdier for at kunne udregne Garrett gennemsnittet. Fra søjle to til seks ses tallet, hvor frekvensen fra tabel B.1 er ganget på den tilhørende Garrett værdi, som ses i tabel B.2. Søjlen "Total" viser den adderet sum og "Garrett Gnm." viser den udregnet gennemsnit. Søjlen yderst til højre viser hvordan de forskellige emner er rangeret.

Som det ses både på figur B.7 og tabel B.4 er *Beskrivelse af formålet* blevet rangeret højst af respondenterne. Dernæst kommer *En oversigt over brugen af biblioteket*. Det ses ydermere at det laveste rangeret emne er *Grafiske elementer*.

Describe in a few sentences what kind of content you would expect to be in user documentation?

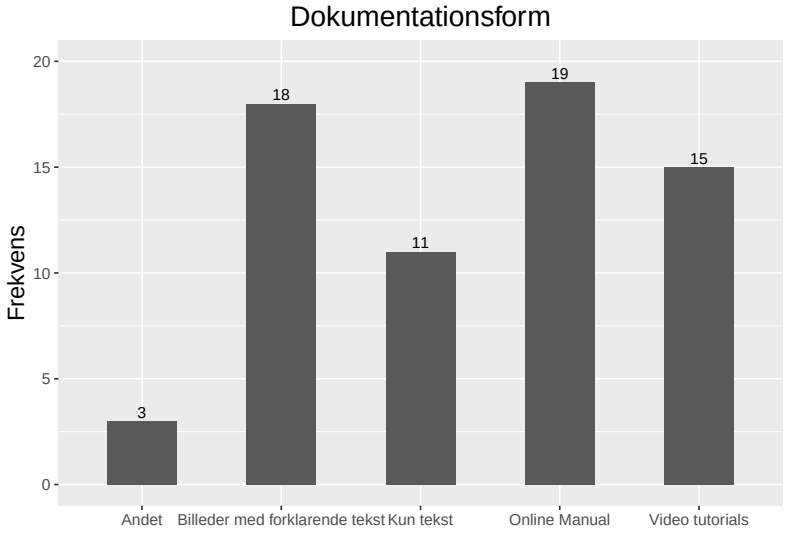


Figur B.8. Figuren viser hvilke indhold respondenterne forventer. Respondenten har kunne skrive op til flere emner og kan derfor være repræsenteret flere steder. Henad x-aksen er emnet og opad y-aksen er frekvensen.

Det ses på figur B.8, at 12 respondenter har nævnt, at de forventer en setup guide, samt kodeeksempler i deres dokumentation, hvilket gør, at det er den mest nævnte. Dernæst kommer "Formål" og "How-to-Guide" som er blevet nævnt hhv. 9 og 10 gange.

Which of the following documentation form do you prefer?

Dette spørgsmål har til formål at undersøge hvilke dokumentationsformer respondenterne fortrækker.



Figur B.9. Figuren viser hvilke dokumentationsform respondenterne fortrækker. Respondenten har haft mulighed for at vælge flere besvarelser. Henad x-aksen er dokumentationsformen og opad y-aksen er frekvensen.

Under kategorien "Andet"er der følgende:

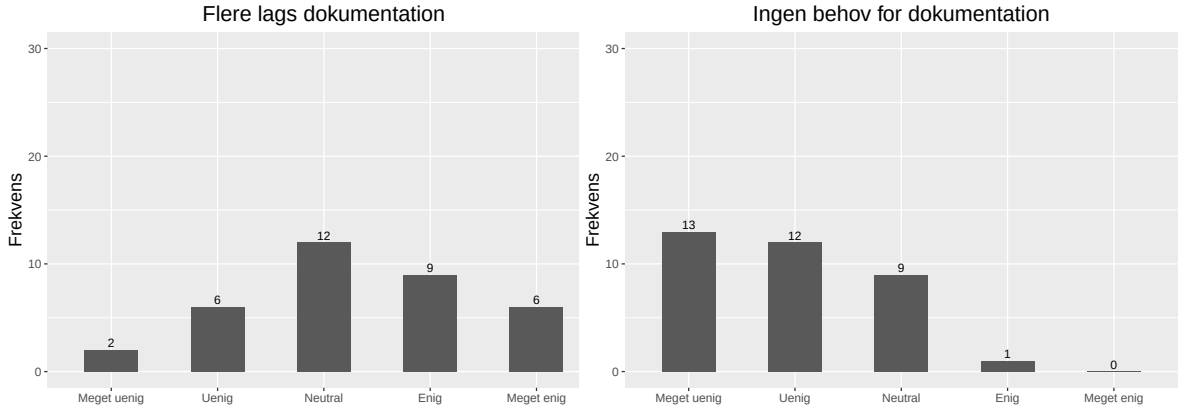
- Interactive (e.g website). It should be easy to search through
- Image examples if the documentation has some visual usecase and videos if thr motion matters, such as a animation library.
- Easy example

På figur B.9 ses det, at online manualler, billeder med forklarende tekst og video tutorials er de tre mest ønskede former for dokumentation.

How much do you agree with the following statements?

De efterfølgende udsagn, som ses på figur B.10, er udarbejdet med det formål at undersøge, hvor enige respondenterne er i udtalelserne sagt under interviewsene, der er beskrevet i afsnit 3.1.

Ifølge Preece et al. [2015][s. 247] bør man bruge fem punkter på en likert skala, når man vil undersøge hvor enige eller uenige respondenterne er i et udsagn. Derfor har respondenten haft netop fem svar muligheder gående fra “Meget uenig” til “Meget enig” i følgende spørgsmål.



(a) Barplot over fordeling af svar på udsagnet: A well documented library has minimum two layers of documentation(...).

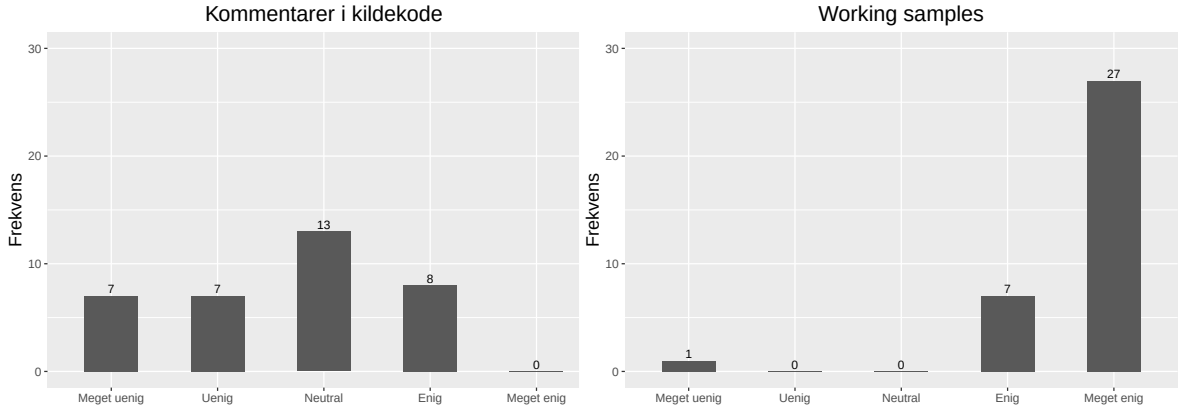
(b) Barplot over fordeling af svar på udsagnet: No need for documentation because good code documents itself.

Figur B.10

Spørgsmålet om flere lags dokumentation, som ses på figur B.10a, er blevet udarbejdet på baggrund af interviewsene, hvor der blev diskuteret om kompleksitet i dokumentationen og hvordan man kunne differentiere dette med flere lag i brugergrænsefladen. Her var en af de interviewedes ideer, at jo dybere man navigerede ned på siden, jo mere kompleks kunne dokumentation blive, da disse oplysninger kun var nødvendige for erfarne udviklere.

Det ses på figur B.10a, at 12 respondenter har forholdt sig neutralt til udsagnet om flere lag af dokumentation, hvilket er den største samling af respondenter. Dernæst kommer "Enig"med 9 respondenter.

På figur B.10b ses det, at 25 respondenter enten har svaret uenig eller meget uenig til, at der ikke er behov for dokumentation, da koden er nok dokumentation i sig selv.



(a) The main part of the documentation should be comments in the source code

(b) A well documented library has working examples

Figur B.11. Barplottene viser fordeling af svar på udsagnet, der står under dem

Ift. om den primære dokumentation bør være kommentarerer i kildekoden, ses det på figur B.11a, at 14 respondenter enten er meget uenige eller uenige. 13 respondenter stiller sig neutrale til dette statement.

34 respondenter er enten meget enige eller enige i påstanden om at et veldokumenteret biblioteket har *working examples*, hvilket ses på figur B.11b.

Which statement do you agree with the most?

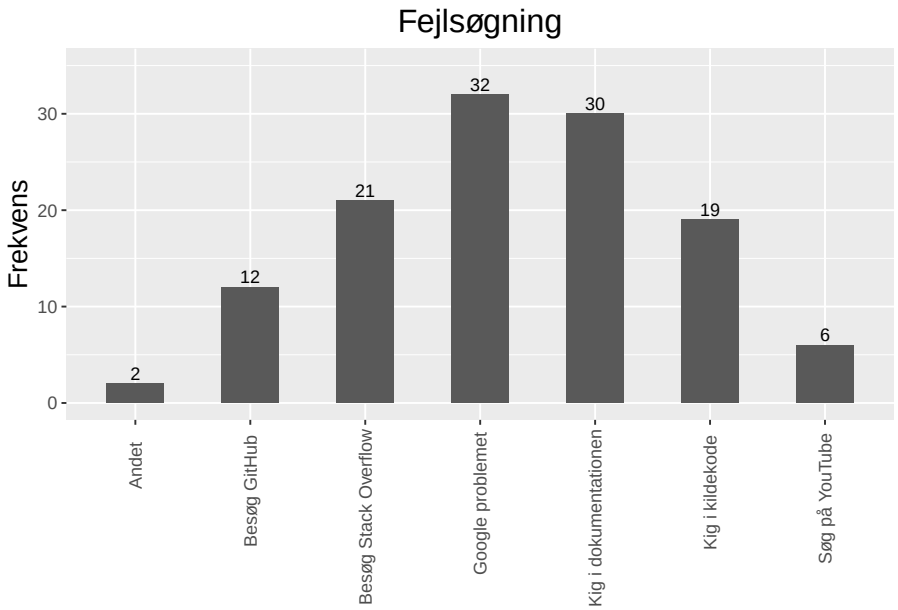
Da resultaterne fra de udførte interviews, som er beskrevet i afsnit 3.1, viste modstridende udtalelser om hvorvidt mængden af dokumentation er en faktor for kvalitet, undersøges dette yderligere med dette spørgsmål.

- A well documented library follow the principle "Less is more": 23
- The more content the documentation has, the better the documentation is: 12

Som det ses på ovenstående tabel har størstedelen af respondenterne været enige med at et veldokumenteret bibliotek følger *Less is more*-princippet.

In general terms what do you do when you get stuck using a third party library?

Dette spørgsmål har til formål, at undersøge hvor respondenterne forsøger at finde svar, når de oplever problemer med det benyttede bibliotek.



Figur B.12. Figuren viser hvad respondenter gør, hvis der opstår en problemstilling med biblioteket. Henad x-aksen står handlingen og opad y-aksen er frekvensen.

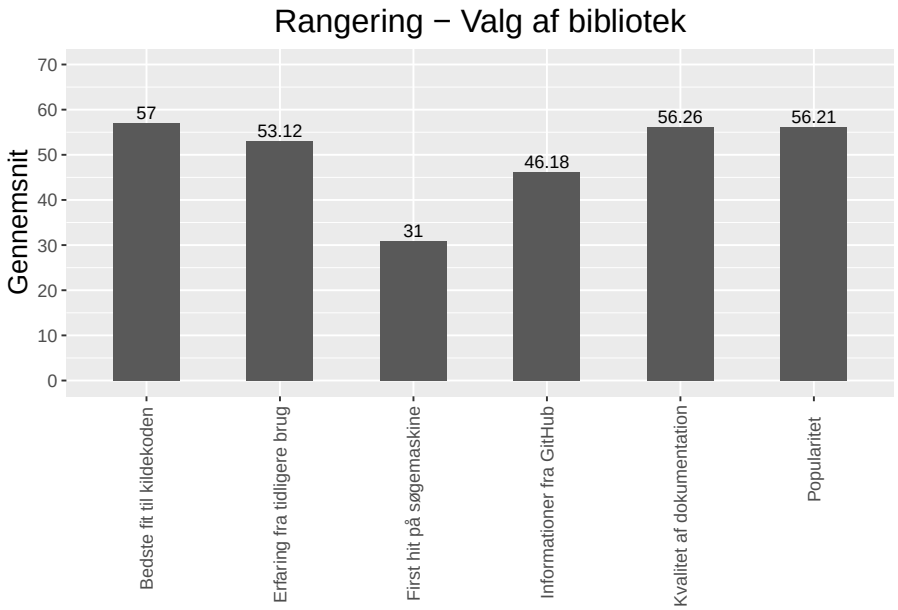
Under “"Andet"” har respondenter skrevet følgende:

- Experiment with the code till I figure it out
- I’ll get advice from my peers if I can

På figur B.12 ses det at de tre primære måder respondenterne fejlsøger på, er ved at benytte en søgemaskine som f.eks. Google, undersøge dokumentationen eller se på hjemmesiden Stack Overflow.

Please rank what you in general find most important when you choose a third-party library

Formålet med dette spørgsmål er, at få indsigt i hvilke faktorer der influerer mest for respondenternes valg af biblioteker. Her har respondenterne ligesom tidligere rangeringsopgaver rangeret forskellige typer udsagn. Udregninger er fortaget på samme måde, som gennemgået i ligning B.1. De eksakte udregninger kan ses i bilag B under filnavnet “Rangeringsopgaver”.



Figur B.13. Figuren viser respondentenes rangering ift. faktorer der influerer i valget af biblioteket. Henad x-aksen står emnet og opad y-aksen er gennemsnittet.

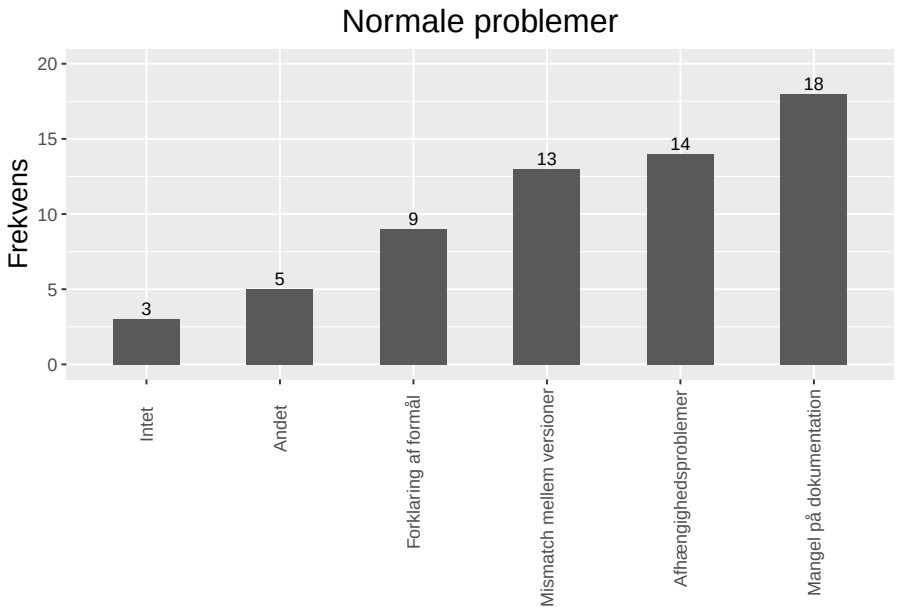
Labels	Rank 1	Rank 2	Rank 3	Rank 4	Rank 5	Rank 6	Total	Garrett Gnm.	Rank
Kvalitet af dokumentation	462	448	432	460	111	0	1913	56,26	2
Popularitet	693	576	108	368	74	92	1911	56,21	3
Informationer fra GitHub	0	512	324	184	481	69	1570	46,18	5
Bedste fit til kildekoden	616	448	432	322	74	46	1938	57	1
Erfaring fra tidligere brug	616	128	540	92	407	23	1806	53,12	4
Første hit på Google	231	64	0	138	69	552	1054	31	6

Tabel B.5. Tabellen viser de nødvendige værdier for at kunne udregne Garrett gennemsnittet. Fra søjle to til seks ses tallet, hvor frekvensen fra tabel B.1 er ganget på den tilhørende Garrett værdi, som ses i tabel B.2. Søjlen "Total" viser den adderet sum og "Garrett Gnm." viser den udregnet gennemsnit. Søjlen yderst til højre viser hvordan de forskellige emner er rangeret.

Som det ses både på figur B.13 og tabel B.5 er *Bedste fit til kildekoden* blevet rangeret højst af respondenterne. Dernæst kommer *Kvalitet af dokumentation*. Det ses ydermere at det laveste rangeret emne er *Første hit på Google*.

Which of the following problems, do you consider normal when using a third-party library?

Dette spørgsmål har til formål at undersøge, hvilke problemstillinger der bliver anset som normale, når man benytter sig af trejdepart biblioteker.



Figur B.14. Figuren viser hvad respondenter mener er normale problemstillinger, når man benytter et trejdepart bibliotek. Henad x-aksen står problemet og opad y-aksen er frekvensen.

Under "Andet"har respondenter skrevet følgende:

- 3rd party framework extension sometimes require extensive change or dont Update when the framework releases new major versions
- I often use the library manager in Processing and Arduino to read about the libraries available
- Sometimes libraries are not fully clear of it's content
- Examples too simple for use, or too complicated
- Outdated documentation.

Som det ses på figur B.14, er de tre mest nævnte problemstillinger mangel på dokumentation, afhængighedsproblemer (oversat fra dependency issues) og mismatch mellem versioner.

If you have any remarks or comments that can help in the development of a intuitive well documented library, please write them belove

Spørgeskemaet blev afsluttet med, at respondenterne havde mulighed for at tilføje ekstra kommentarer som de mente vil kunne hjælpe til at opnå projektets formål. Dette var valgt for at forsøge, at åbne for ideer fra respondenterne på et mere bredt plan end resten af spørgeskemaet giver mulighed for.

- No need for documentation because good code documents itself Is only true for internal projects. And is the first line of documentation. Third party code needs way more documentation. Ruby on Rails and Ruby has the best documentation in the world. https://ruby-doc.org/core-3.1.1/Enumerable.html#method-i-tally
- Most major libraries within c sharp tend to have appropriate documentation.

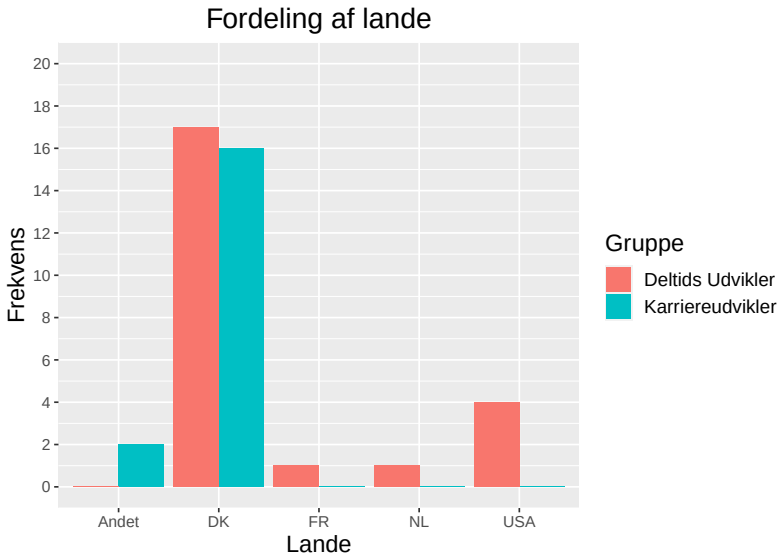
- Look at other projects and see What You want to do nettet but also What they do good
- Take inspiration from vue documents. It might be a big framework, but their documentation is one of the best on the web.
- When writing documentation, remember that people has different skill sets and different experience.Do an in depth and a TLDR section of the documentation.Good headline Brief summary of what the code does- The TLDR for experiencedn programmers. The in depth for people wanting to know more or learn how it works in details
- Too many developers use undefined acronyms about which we have to guess. In the Arduino world the use of delay() is far too often in examples making them nearly useless for integration with other code and constantly reenfources poor thinking about program structure.Note this editing environment seams to have disabled the browser spelling help. I can not right click and get suggestions on the words where bad spelling is indicated with the red underline. Ironic is it not.
- Update, update, update. Most libraries I come across tend to be outdated

B.2 Svarfordeling for Deltidsudviklere og Karriereudvikler

Følgende afsnit viser de demografiske forskelle, der er i de to grupper "Karriererudvikler" og "Deltids Udvikler".

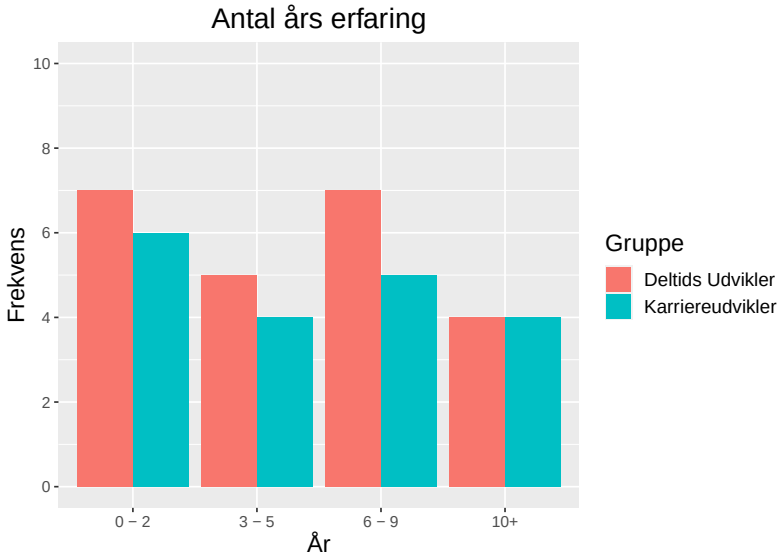
Figur B.15. Figuren viser aldersfordelingen af de to grupper. Henad x-aksen vises aldersgrupper og opad y-aksen vises frekvensen.

På figur B.15 ses det, at det gælder for begge grupper at over 75% af respondenterne er imellem 20 til 39 år. De er ydermere fem deltidsudviklere der er over 49 år. Der er en respondent af karriereudviklere, der er under 20 år og en respondent der er imellem 40 og 49 år.



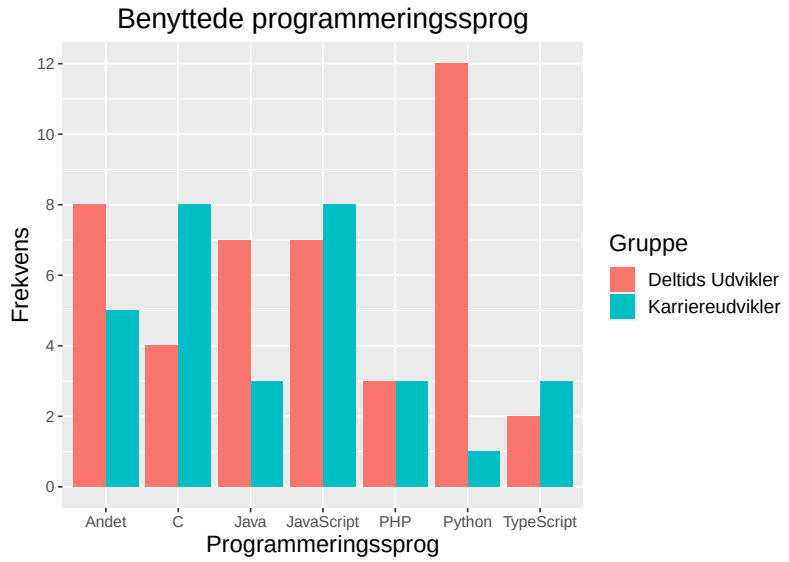
Figur B.16. Figuren viser hvordan respondenterne er spredt i hvor de er bosiddende. Henad x-aksen vises nationalitet og opad y-aksen vises frekvensen.

Som det ses på figur B.16 gælder det for begge grupper, at over 70% af respondenterne er bosiddende i Danmark. I gruppen "Karriereudvikler" har to ikke ønsket at udtale sig om dette. To respondenter i gruppen "Deltids Udvikler"er bosiddende enten i Frankrig eller Holland. I samme gruppe er godt fire respondenter bosiddende i USA.



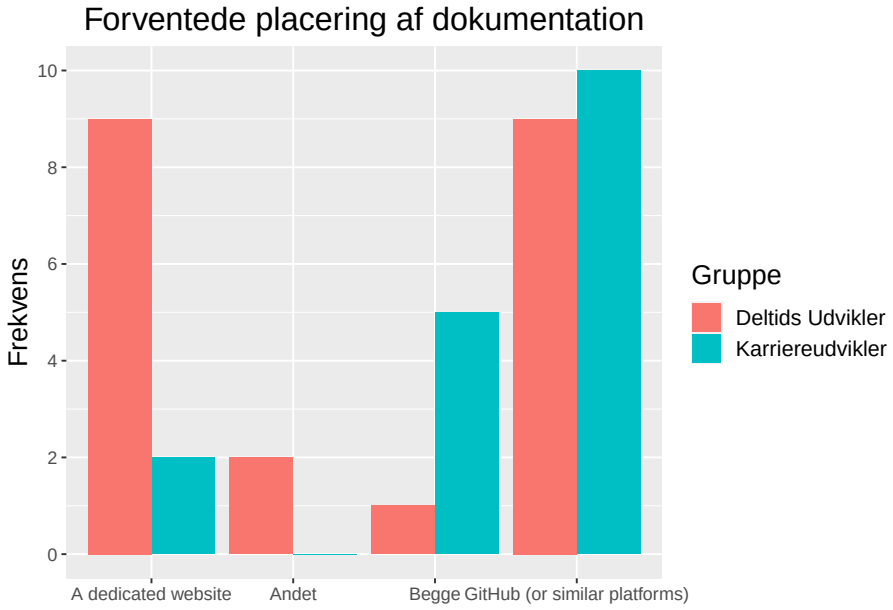
Figur B.17. Figuren viser, hvor mange års erfaring respondenterne har. Henad x-aksen vises aldersgrupperne og opad y-aksen vises frekvensen.

På figur B.17 ses det, at begge grupper har respondenter, der repræsenteres i alle intervaller med fire repsondenter eller derover. For "Deltids Udvikler" er der flest respondenter samlet i 6 - 9 års erfaring og 0 - 2 års erfaring med syv respondenter i hver. For "Karriereudvikler" er der flest respondenter repræsenteret i intervallet 0 - 2 års erfaring med seks respondenter.



Figur B.18. Figuren viser, hvor mange års erfaring respondenterne har. Henad x-aksen vises aldersgrupperne og opad y-aksen vises frekvensen. Respondenterne har haft mulighed for at svare mere end et programmeringssprog og kan derfor være repræsenteret i flere sprog. Grafen er derfor et udtryk for fordelingen ift. alle de programmeringssprog der er nævnt.

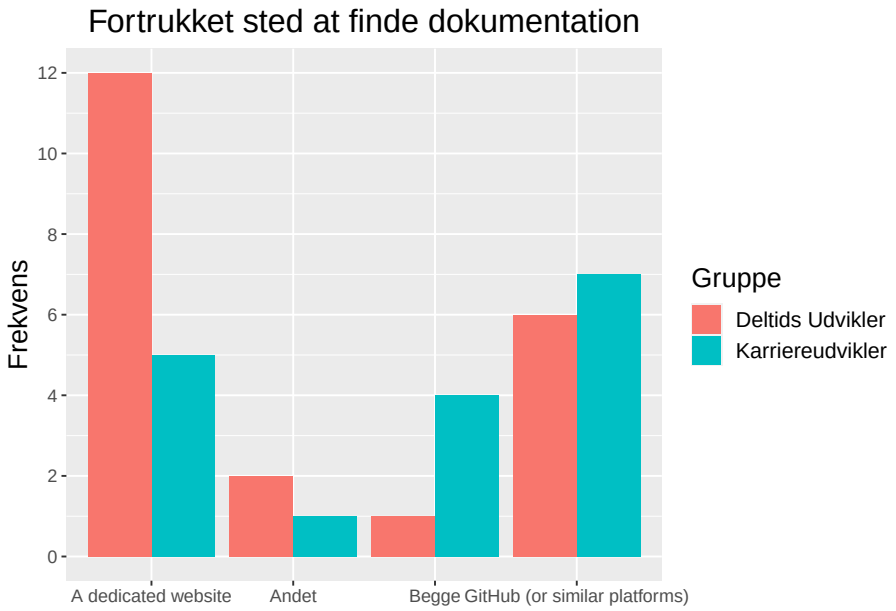
Som det ses på figur B.18 er Python det programmeringssprog som flest respondenter i gruppen "Deltids Udvikler" benytter, da 12 af respondenter bruger dette. Gruppen "Karriererudvikler" benytter sig primært af C og JavaScript med cirka otte respondenter repræsenteret i hver programmeringssprog.



Figur B.19. Figuren viser fordelingen af de to grupper "Deltids Udvikler" og "Karriereudvikler" ift. forventet placering af dokumentation. Henad x-aksen er de valgte emner. Opad y-aksen er frekvens repræsenteret.

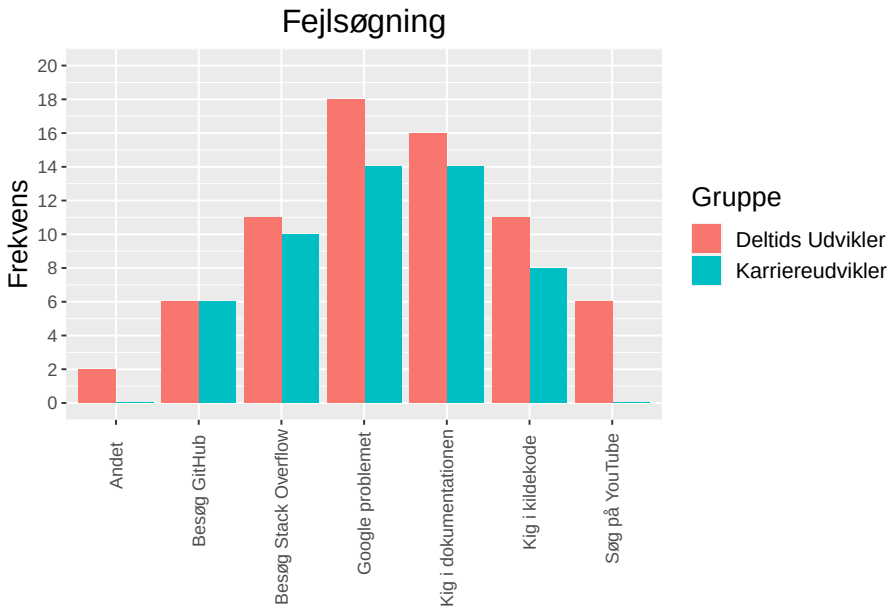
På figur B.19 ses det, at ti respondenter i gruppen "Karriererudvikler" forventer at finde

dokumentation på GitHub eller lignede platforme, hvilket gør det til det primære sted for denne gruppe. Gruppen "Deltids Udvikler" forventer primært at finde dokumentationen på en dedikeret hjemmeside.



Figur B.20. Figuren viser fordelingen af de to grupper "Deltids Udvikler" og "Karriereudvikler" ift. fortrukket placering af dokumentation. Henad x-aksen er de valgte emner. Opad y-aksen er frekvens repræsenteret.

Det ses på figur B.20, at ligesom tilfældet var ved forventet placering af dokumentation fortrækker "Karriereudvikler" primært at finde dokumentationen på GitHub eller lignede platforme: Gruppen "Deltids Udvikler" fortrækker derimod primært at finde dokumentationen på en dedikeret hjemmeside.



Figur B.21. Figuren viser fordelingen af de to grupper "Deltids Udvikler" og "Karriereudvikler" ift. hvordan de fejlsøger. Henad x-aksen er de valgte emner. Opad y-aksen er frekvensen repræsenteret.

På figur B.21 ses det at begge grupper primært søger på deres problem på Google eller lignede søgemaskine. Sekundært, undersøger begge grupper dokumentationen for informationer.

Eksakte værdier for rangeringsopgaver

Følgende afsnit viser de eksakte værdier fra rangeringsopgaverne i spørgeskemaet.

Fælles for alle tabeller i dette afsnit er, at søjler med “Rank” viser værdien for udregningen forklaret ved ligning B.1. Søjlen "Total" er den totale værdi for emnet, hvor de vægtet værdier er adderet sammen. Søjlen "Garrett Gnm." viser den eksakte gennemsnit værdi for emnet. Rank-søjlen viser hvilke rangering det pågældende emne er rangeret til.

Labels	Rank 1	Rank 2	Rank 3	Rank 4	Rank 5	Total	Garrett Gnm.	Rank
Nem setup	375	240	200	200	48	1063	53,15	2
Worked-examples/Kode snippets	375	480	100	160	24	1139	56,95	1
Dybdegående dokumentation	150	120	400	160	96	926	46,3	4
Komplette eksempler	450	120	200	160	96	1026	51,3	3
Dokumentation i kildekode	150	240	100	120	216	826	41,3	5

Tabel B.6. Tabellen viser resultaterne for gruppen "Deltids Udvikler".

Labels	Rank 1	Rank 2	Rank 3	Rank 4	Rank 5	Total	Garrett Gnm.	Rank
Nem setup	525	300	100	120	0	1045	61,47059	2
Worked-examples/Kode snippets	525	240	300	0	0	1065	62,64706	1
Dybdegående dokumentation	0	240	150	160	144	694	40,82353	4
Komplette eksempler	75	180	250	280	24	809	47,58824	3
Dokumentation i kildekode	150	60	50	120	240	620	36,47059	5

Tabel B.7. Tabellen viser resultaterne for gruppen "Karriereudvikler".

Tabel B.6 og B.7 viser de eksakte værdier for spørgsmålet "Please rank what you in general find most important for documentation for a library"

Labels	Rank 1	Rank 2	Rank 3	Rank 4	Rank 5	Total	Garrett Gnm.	Rank
Overblik over bibliotekets flow	75	180	350	160	120	885	44,25	4
En oversigt over brugen af biblioteket	225	540	150	160	24	1099	54,95	2
Beskrivelse af formålet	600	60	350	80	48	1138	56,9	1
Grafiske elementer	375	0	100	200	192	867	43,35	5
Nem navigation	225	420	50	200	96	991	49,55	3

Tabel B.8. Tabellen viser resultaterne for gruppen "Deltids Udvikler".

Labels	Rank 1	Rank 2	Rank 3	Rank 4	Rank 5	Total	Garrett Gnm.	Rank
Overblik over bibliotekets flow	150	240	50	280	48	768	48	4
En oversigt over brugen af biblioteket	300	180	250	120	24	874	54,625	2
Beskrivelse af formålet	375	300	250	40	0	965	60,3125	1
Grafiske elementer	0	60	150	40	264	514	32,125	5
Nem navigation	375	180	100	160	48	863	53,9375	3

Tabel B.9. Tabellen viser resultaterne for gruppen "Karriereudvikler".

Tabel B.8 og B.9 viser de eksakte værdier for spørgsmålet "Please rank what you in general find most important for documentation for a library"

Labels	Rank 1	Rank 2	Rank 3	Rank 4	Rank 5	Rank 6	Total	Garrett Gnm.	Rank
Kvalitet af dokumentation	231	256	270	276	37	0	1070	56,31579	1
Popularitet	385	384	54	138	37	69	1067	56,15789	2
Informationer fra GitHub	0	128	270	138	333	0	869	45,73684	5
Bedste fit til kildekoden	308	320	162	184	37	46	1057	55,63158	3
Erfaring fra tidligere brug	308	128	270	46	222	23	997	52,47368	4
First hit på søgemaskine	192	0	0	74	23	299	588	30,94737	6

Tabel B.10. Tabellen viser resultaterne for gruppen "Deltids Udvikler".

Labels	Rank 1	Rank 2	Rank 3	Rank 4	Rank 5	Rank 6	Total	Garrett Gnm.	Rank
Kvalitet af dokumentation	231	192	162	184	74	0	843	56,2	3
Popularitet	308	192	54	230	37	23	844	56,26667	2
Informationer fra GitHub	0	384	54	46	148	69	701	46,73333	5
Bedste fit til kildekoden	308	128	270	138	37	0	881	58,73333	1
Erfaring fra tidligere brug	308	0	270	46	185	0	809	53,93333	4
First hit på søgemaskine	0	54	0	37	46	253	390	26	6

Tabel B.11. Tabellen viser resultaterne for gruppen "Karriereudvikler".

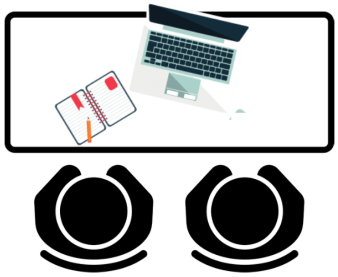
Tabel B.10 og B.11 viser de eksakte værdier for spørgsmålet "Please rank what you in general find most important when you choose a third-party library".
Ved denne rangerginsopgave har respondenterne haft seks emner de skulle rangerer og derfor er der tilføjet en ekstra søjle til tabellen og udregningen af værdierne har gjort at der skulle tages seks værdier med. Udregningen for tabellen kan ses i bilag A i mappen "Adskilt".

Heuristisk Evaluering C

Dette kapitel har til formål at redegøre for et uddybende referat af den heuristiske evaluering. Rådata såsom optagelser og noter kan findes i bilag C.

C.1 Forsøgsopstilling

Evalueringen fandt sted i et mødelokale ved Seluxit.



Figur C.1. Figuren viser forsøgsoptillingen under evalueringen, hvor evaluatoren og facilitatoren er placeret på samme side af et bord og har en computer og noteblok foran sig.

Som vist på figur C.1 er facilicator og den interviewede placeret ved samme bord. Foran dem er der placeret en computer og papirblok. Bag dem er et whiteboard, som er tiltænkt hvis den interviewede fortrækker at forklare og skitser via brug af denne. Evalueringen forgår på engelsk.

C.2 Referat

Evalueringen blev indledt med en forklaring af projektets formål og fokuspunkter, samt hvordan evalueringen vil foregå.

Dernæst præsentere den interviewede sig selv med fokus på arbejde og uddannelsesbaggrund. Her forklarede den interviwede at hun hedder Evelina Mihailova og er teamlead for UX/UI ved Seluxit A/S. Hun har en bachelor indenfor web udvikling og grafisk design. Hun har ydermere taget en kandidat uddannelse i *Information Architecture* på Aalborg Universitet. I den resterende del af referatet vil den interviewede blive benævnt ud fra hendes fornavn.

Intervieweren forklarede herefter, hvordan evalueringen vil foregå. Her blev det forklaret at evalueringen er delt op i to faser, hvor Evelina til at starte med vil få til opgave at navigere tilfældigt rundt i prototype og udtrykke sine umiddelbare tanker. I fase 2 af evalueringen

vil siderne i prototypen derimod blive gennemgået med fokus på Jakob Nielsen [1994] ti design heuristikker.

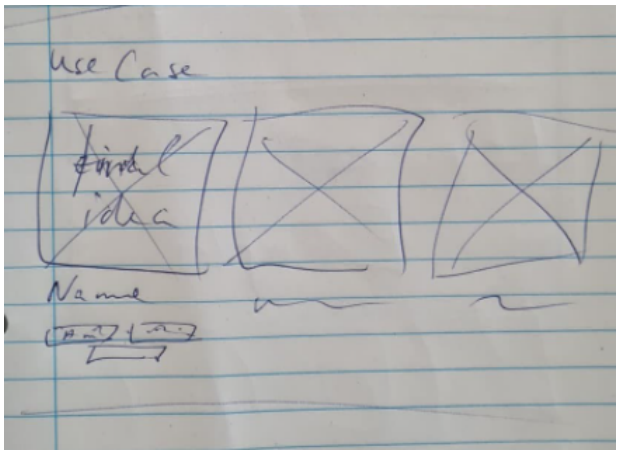
Herefter sættes den første fase igang, hvor prototypen er vist og udgangspunktet til at starte med er forsiden.

Det første Evelina bemærker er, at hun bliver forvirret over konteksten. Dette begrunder hun med, at Seluxit logoet er blandet ind i et biblioteket for Wappsto. Dette kan ifølge Evelina skabe forvirring, da Seluxit er virksomheden og wappsto er platformen, så forsiden minder for meget om en forside til hele Seluxit.
Hun forklarer, at hvis hun søgte på wappsto dokumentation, vil hun ikke forvente at komme til en hjemmeside som denne. Dette er grundet, at hun vil forvente at komme til en side, hvor der er API dokumentation og andre programmeringssprog. Hun eksemplificere dette ved at nævne, at hvis hun søgte på Wappsto Arduino, så vil hun finde noget specifikt til Arduino og på samme måde, hvis hun søgte på Wappsto JavaScript IoT, så vil hun komme hen et sted, hvor der var formidlet informationer om et bibliotek til JavaScript. Hun udtaler ydermere, at hun også vil forvente at komme til et sted hvor det *keyword* hun har brugt er vist. Evelina begynder dernæst at nærlæse introduktions teksten og forklare, at nu når man beskriver specifikke sprog i introduktionen vil hun forvente en specifik retning for hvert programmeringsprog.

Evelina begynder herefter at scrolle ned igennem forsiden og ved sektionen "Community", forklarer hun at man sjældent ser nogle designe en dropdown menu ind på et ikon, som det er gjort i prototypen. Evelina forklarer, at hun vil forvente at komme direkte til Github hjemmesiden ved at trykke på ikonet. Fra GitHub vil man så skulle vælge hvilket programmeringssprog, der er relevant for en. Hun forslår, at man ændre dette til, at designe en sektion kun med Github, hvor man kan se logoerne med Arduino, Python og JavaScript.

Ydermere forklarer hun, at hun ikke vil forvente at man kan navigere direkte til Stack Overflow, da man som regel kun vil gå dertil hvis man har problemer med biblioteket.
Her bliver det dog forklaret at Stack Overflow giver mulighed for at følge tags og derfor vil man via denne knap tilgå de problemstillinger som andre i miljøet har haft. Her udtrykker Evelina bekymring over, at Wappsto ikke har et stort nok miljø lige nu til denne funktion.

Dernæst scroller Evelina videre ned og ser "Example" sektionen og går derefter op igen til Use case sektionen. Her forklarer Evelina, at de ligner meget hinanden og spørger ind til om de ikke vil gøre det samme. Til det bliver der forklaret, at Use Case er mere hele projekter, hvor eksempler er små konkrete eksempler, som man kan kopier og sætte over i sin kode. Evelina forklarer hertil, at *Use cases* og *Example* på nuværende stadie ligner meget hindanden og for at undgå forvirring vil hun vil designe dem forskelligt.

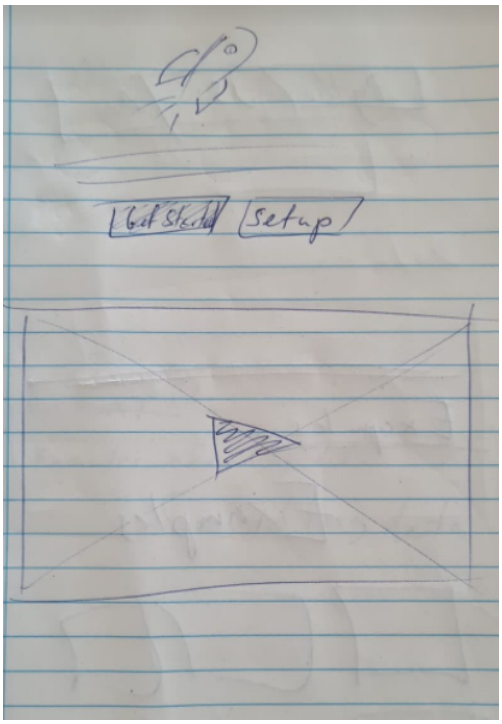


Figur C.2. Figuren viser evaluatorens skitse af et eksempel på hvordan man kunne opstille links til GitHub på forsiden.

Til dette forslår Evelina et alternativ, som ses på figur C.2. I stedet for et billede af koden, vil hun have et billede af det endelig produkt og så en kort introduktion tekst med titel. Hun tilføjer ydermere, at man kunne designe for små tags, hvor brugeren kan se hvilke programmeringssprog use casen er kompatibel med.

Dernæst scroller Evelina videre ned til "get started" nederst på forsiden. Her forklarer Evelina, at dette vil hun forvente at se i toppen af forsiden. Hertil forklarer hun, at videoer som introduktionsvideoen på forsiden som regel vil være placeret på en marketingside og ikke i dokumentationen for et bibliotek.
Hun uddyber dog, at det godt kunne virke, men hun mener det er svært at vurdere, da hun ikke har set indholdet af videoen endnu.

Der bliver ydermere forklaret, at "Get started" knappen er gemt væk, da fokus vil være på videoen og det besværliggøre at finde knappen for at starte med at bruge biblioteket. Derfor vil hun fortrække, at man flyttede designet med "Get started" fra nederst på forsiden til øverst på siden, så man lægger mærke til, hvor man kan starte hurtigere. Til det foreslår hun en knappe, der hedder "Get started", hvor man får en introduktion og brugeren bliver guidet igennem biblioteket og en der hedder "setup", hvor man går direkte til opsætningen af biblioteket. Dette bliver begrundet med, at nogle personer er ligeglade med introduktion og vil bare have sat biblioteket op. Yderligere bliver der forklaret at man ikke ved om det er en begynder eller erfaren person, der skal benytte systemet og den erfarne har ikke lyst til at skulle gennemlæse hele introduktion, men vil fortrække at skulle gå direkte til opsætningen, hvor man kan kopier koden og opsætningen hurtigt over i egen kildekode. Dette er grundet, at man som erfaren bruger ved hvad man vil og hvad man skal bruge det til, da der er en grund til at man er inde på Wappsto og WappstoIoT.



Figur C.3. Figuren viser evaluatorens skitse af et eksempel på hvordan man kunne opstille "Get Started" og "Setup" på forsiden.

Dernæst eksemplificere Evelina sin forklaring med tegningen, der vist på figur C.3. Hertil bliver det forklaret, at det er fint at have et ikon i toppen og en introduktionstekst, men man bør fremhæve "Get started" for at tilgodese den uerfarne i domænet.
Til spørgsmålet om hvad Evelina vil forvente hvis man trykkede på "Get started " knappen forklarer hun, at det kommer an på konteksten, normalt vil man forvente i "get started" en mere teknisk forklaring af, hvordan biblioteket er opbygget. Den nuværende introduktion øverst på forsiden er for hende marketing, da den forklarer mere, hvorfor man skal bruge dette biblioteket over et andet.

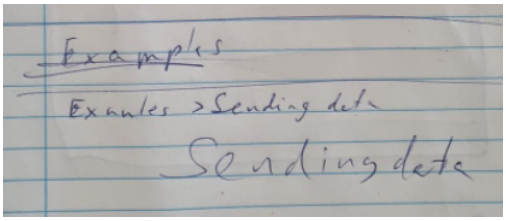
Dernæst trykkes der på "Get started" knappen. Her trykkes der frem og tilbage imellem stepsene. Til dette forklarer Evelina, at hun vil forvente et step før dette hvor man skulle vælge sit fortrukket programmeringssprog. Dette skulle være for at undgå, at man skal vælge programmeringssprog hver gang man tilgår en fane eller side. Dette valg forklarer hun, at det kan blive fortaget på forsiden og så blive gemt, når man navigere rundt i dokumentationen. Et alternativ til dette vil være, at placere valget af programmeringssprog i starten af opsætningen. Evelina fremhæver i denne kontekst dokumentationen for Wappsto Native[1], hvor man i opsætningen bliver spurgt ind til om man ønsker at fortage opsætningen til Windows eller Mac og om man vil tilslutte en mobil. Ud fra de beslutninger navigere man så kun videre i den kontekst.På denne måde undgår man, at man kopier noget forkert over til sit projekt.

Ved Step 1 i opsætningen udtaler Evelina, at der ikke er nogen grund til at lave så have så mange detaljer for at opsætte en konto fordi det vil være naturligt for udviklere at gøre. Dertil foreslår Evelina, at et hyperlink og forklaring vil være fyldestgørende.

[1] `https://github.com/Wappsto/rn-wappsto-networks`

Det næste Evelina bemærker er, at der er stor forskel i detaljeringsgraden imellem step 1 i opsætningsguiden, hvor man skal oprette en konto og til at man skal hente et certifikat i Wappsto. Her forklarer Evelina, at i en "Get started" guide vil man forvente at blive guidet til en side, hvor brugeren får forklaret hvad er det man gør og hvorfor gør man det. Som det er i prototypen på nuværende stadie bliver konteksten og formålet ikke forklaret med certifikatet Ydermere forklarer Evelina til opsætningen, at det er fint at have et *first example*, men det bør mere være en del af "Get Started" guiden.

Til siden "Sending data" bliver der forklaret, at knapper til at vælge programmeringssprog øverst på siden ligner mere tags end knapper. Dette bliver begrundet med, at de er for lille og bør blive forstørret. Dernæst navigere Evelina lidt rundt i prototypen igen og forklarer, at "Examples" siden er som forventet, men steppet fra den side og til "Sending data" siden er forvirrende. Dette bliver uddybet med, at der mangler en måde at komme tilbage på eller at man kan se eksemplerne via en sidebar, så man kan navigere sig mere frit rundt uden at skulle gå frem og tilbage i dokumentationen hele tiden.



Figur C.4. Figuren viser evaluatorens skitse af et eksempel på hvordan man lave en breadcrumnp på Example-siden.

Til spørgsmålet om Evelina forventede at komme til et "Working example" fra "Examples" siden, blev det forklaret, at det ville hun, men hun vil også forvente at kunne se *breadcrumbs*. På figur C.4 ses, hvorledes Evelina eksemplificere denne forventning.

Det er dog ikke klart i det nuværende design, at man er på en side for working example, da titlen er den samme som på forrige side, hvilket forvirrende Evelina.

Under punktet "What you need" på "Sending data"-siden beskriver Evelina, at grundet de forskellige kategorier står ved siden af hinanden ligner det mere, at man enten skal bruge det ene eller det andet. Her bør man istedet sætte det op som i en vertikal punktform. Hyperlinket "Complete code" kan virke forvirrende ifølge Evelina, da den står ved introduktionen og man vil forvente at den går til en ny side. Her bliver det forklaret, at den heller ikke bør stå under en overskrift. Dette bliver begrundet med, at det ikke ofte at man vil kopier en kode man ikke har set endnu. Evelina udtaler ydermere, at det er ikke tit at man har en video i eksempler man plejer blot at have en kode man kan køre og se. Ydermere forklarer Evelina, at hun vil fortrække at man bare skrev en introduktion i stedet for at fortælle om det i en video.

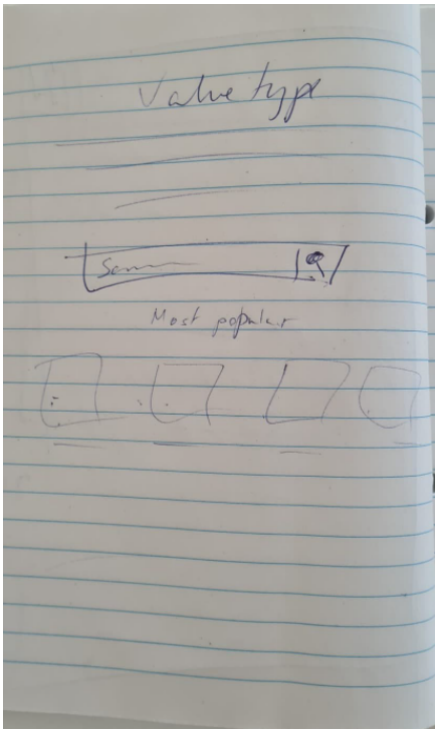
Dernæst scroller Evelina ned til bunden af siden. Her bliver det bemærket, at det er godt at man kan se hvad der sker i Wappsto, når man har lavet et eksempel. En mulighed tilføjelse til dette, kan dog være at tilføje et link til Wappsto, så man ikke behøver at navigere manuelt til Wappsto hjemmesiden.

Dernæst navigere Evelina til siden "Value types". Her stilles der spørgsmål ved titlen,

da hun er usikker på om det er et signede navn for udefrakommende personer. Som introduktion til siden "Value types", bliver det forslået, at der bliver forklaret hvad "value types" er og hvad det bruges til. Dette skal være med til, at gøre det lettere for brugeren at få en forståelse for om dette er relevant at benytte for brugeren. Evelina forslår ydermere, at tilføje et link til datamodellen.

Evelina forsætter derefter til "Sending data" siden. Her forklarer hun at man ikke bør opdele koden i mindre dele, da det vil besværliggøre brugen af koden, da brugeren som regel gerne vil kopier koden direkte over i eget projekt og se det virke. I stedet forklarer Evelina at man kunne adskille "Sending data" siden, så den venstre side vil være forklaring af koden med tilhørende billeder og højre side vil den samlet kode være.
Ift. den venstre side, fastslår Evelina at man kun bør adskille koden de steder, hvor man vil kunne kopier koden direkte over uden den resterende del og få koden til at virke uden videre.

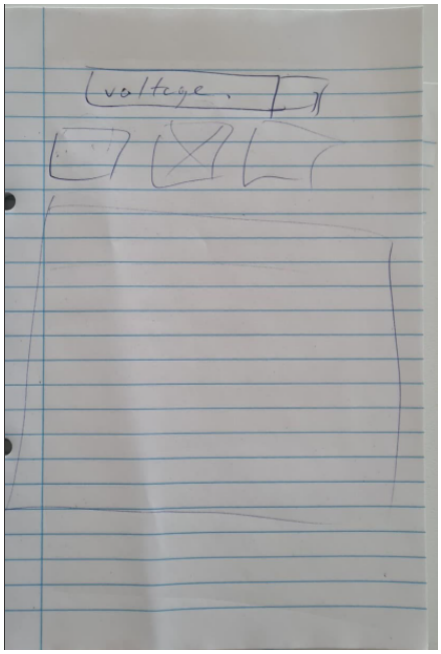
Dernæst navigere Evelina sig videre til "Voltage measurments" eksemplet. Her forklarer Evelina igen, at for at øge den nemme navigation, kunne man tilføje "breadcrumbs" på den side, så man har mulighed for at bevæge sig frem og tilbage på en nem og intuitiv måde. Til dette påpeger interviewer, at der er designet for at brugeren kan navigere imellem kategorierne via en sidebar. Denne sidebar havde Evenlina dog ikke set og kommentere, at hun normalt vil designe en sådan side til at fylde hele den ene side, så man nemmere, som bruger, lægger mærke til den. Det næste Evelina kommenterer på, er grupperingen er "Value types". For at kunne designe for, at der i fremtiden kan være mange flere typer og derved kategorier af "value types", kan det nuværende design hurtigt blive uoverskueligt. Man kunne derfor allerede nu designe for en mere avanceret søgefunktion.



Figur C.5. Figuren viser evaluatorens skitse af et eksempel på hvordan man opsætte en anvanceret søgefunktion på "Value types siden.

Dette bliver eksemplificeret med skitsen på figur C.5. Skitsen bliver forklaret med, at der først bør være en introduktion på siden, og dernæst en søgefunktion hvor man søge på navn eller SI enhed. Dernæst kunne man have de mest populære kategorier, som brugeren kan få inspiration af.

Dernæst trykkes der ind på "Voltage measurements" og det bliver forklaret, at man godt kunne sammenflette de sider "Voltages types" og "Voltage measurements" for at gøre navigeringen nemmere.



Figur C.6. Figuren viser evaluatorens skitse af et eksempel på hvordan man opsætte en anvanceret søgefunktion på "Value types" siden.

Dette bliver eksemplificeret med skitsen på figur C.6. Hvor man kunne søge på voltage og så kommer der forskellige kategorier op og dernæst vælger man som bruger, hvilken en man ønsker og så kommer værdierne frem.

Ift. til den tabel der vises, forklarer Evelina at nogle vil fortrække ikoner frem for SI enheder.

Dernæst navigere Evelina sig tilbage til "Value types" siden. Her forklares det, at grupperne af "value types" burde være mere konkrete f.eks. så er "climate types" for abstrakt og bør være mere konkret såsom "Temperature".

Det næste Evelina kommentere på, er at man bør erstatte de eksempler der kommer efter tabellen, med en søjle mere i tabellen, hvor man kan trykke og se eksemplet. Dette er for at gøre siden mere simpelt. Dernæst forklare Evelina også at det nødvendigvis ikke behøves at lave en ny side, men man kan nøjes med et popup billede, hvor eksemplet bliver forklaret.

Dernæst navigeres der tilbage til "getting started" siden. På denne side forklarer Evelina at denne side burde indeholde links til datamodellen og "value types". Her forslås det også, at man kunne tilføjes en sidebar, hvor man har hele "Getting started" stepsense vist, så man kan navigere direkte til specifikke punkter

Dernæst navigere Evelina sig videre til forsiden, hvor det bliver positivt bemærket, at der står hvilken version dokumentationen omhandler. Der forslås, at man kan tilføje en dropdown menu, så man kan vælge tidligere versioner. Man behøver dog nødvendigvis ikke at skelne imellem de mindre versionsopdateringer, da der sjældent er større ændringer, der har indflydelse i disse.

Efter denne feedback, forslår intervieweren at der forsættes til fase 2 af evalueringen. Her indledes der med en kort introduktion af hvordan den fase af evalueringen kommer til at forgå. Evelina responderer dog ved, at hun mener at det vil være en gentagelse af det allerede sagte og at hun finder det svært at gennemgå prototypen ud fra heuristikkerne grundet prototypens tidlige stadie.

Dog gennemgår hun hvert heuristik på et overordnet plan. Her forklarer hun, at hun ikke har noget at tilføje til "Visibility of system status", "Flexibility and efficiency of use" og "Aesthetic and minimalist design" og at hun ikke kan give feedback på "Match between system and the real world", "Error prevention", da prototypen er på et for tidligt stadie. "User control and freedom" bliver eksemplificeret ved at det handler om at brugeren ikke skal bruge for mange klik for at nå sit mål. Ift. til "Recognition rather than recall", nævner Evelina, at dropdown menuen og burger menuen bør kunne ses hele tiden for at opnå denne heuristik.
Til sidst tilføjer hun, at hun ikke mener at "Help users recognize, diagnose, and recover from errors", samt "Help and documentation" er rette heuristikker til denne type dokumentation, så der har hun heller ikke noget at tilføje.

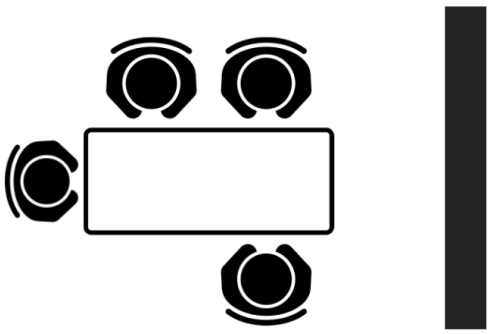
Herefter afsluttes evalueringen og der slukkes for optagelserne.

Feedback fra udviklere D

Dette kapitel har til formål at beskrive fremgangsmåden for en feedback session, der blev afholdt med tre udviklere. Ydermere har kapitlet til formål, at beskrive resultaterne fra denne session i form af et referat. Rådata såsom optagelser, PowerPoint præsentation og noter, kan findes i bilag E.

D.1 Forsøgsopstilling

Feedback mødet fandt sted i et mødelokale ved Seluxit.



Figur D.1. Figuren viser forsøgsoptillingen under feedback mødet. Personen yderst til venstre er facilitatoren. De resterende personer er deltagerne. Firkanten yderst til højre er et projektor lærred, hvor PowerPoint præsentationen kunne ses.

Som vist på figur D.1 er facilicator placeret ved enden af bordet, hvor de deltagende sidder på hver sin side af bordet. For at fremvise resultaterne fra tidligere forsøg, samt andet relevant indhold, er der udarbejdet et PowerPoint præsentation. Dette vil blive vist på lærred, som er skitseret som en sort firkant yderst til højre på billedet.

D.2 Referat fra feedback mødet

Følgende referat omhandler et feedback møde med tre udviklere fra Seluxit. Alle tre udviklere har deltaget i de tidligere gennemførte semi strukturede interviews (se afsnit 3.1.1 for yderligere demografiske oplysninger). Deltagernes navne er Andreas, Karsten og Michael.

Til afholdelse af mødet ageret projektmedlemmet facilitator. Herudover blev mødet videooptaget, og den fulde optagelse af mødet kan findes i bilag E.

Mødet blev indledt med en agenda, hvorefter der blevet gennemgået en kort introduktion til formålet med projektet og hvilke fokuspunkter projektet har. Dernæst blev tidligere resultater, samt andet relevant information gennemgået. Herefter blev hele prototypen gennemgået side for side af facilitatoren. I denne del af mødet, blev der ikke givet feedback på prototypen.

Efter prototypen var gennemgået, navigerede facilitatoren tilbage til forsiden på prototypen og spørger indledningsvis indtil deltagernes tanker om denne side. Her indleder Andreas med at sige, at versionsnummeret, som er placeret øverst i midten af forsiden, bør flyttes ud i venstre side. Dette begrunder Andreas med at man altid bør kunne se hvilken version man kigger på, når man navigerer sig rundt i dokumentationen. Til dette spørger facilitatoren ind til placeringen af denne information og der bliver svaret enstemmig fra gruppen, at den bør være i toppen af sidemenuen. Andreas uddyber sin forklaring med, at hvis man vælger en ældre version, så burde der være en advarsel på siden. Dette er grundet, at man hele tiden som bruger ønsker at kunne se hvilken version af dokumentationen man kigger på. Michael forklarer herefter, at et problem ved denne version nummerering er, at de forskellige sprog ikke følger hinanden. Dog tilføjer Karsten, at man følger hinanden i de store udgivelser, men det kun er de mindre opdateringer de ikke følger hinanden. Herefter forklarer Andreas, at han regner med at versionsnummeret, der er designet for i prototypen omhandler versionsnummeret for dokumentationen og ikke selve biblioteket. Her forklarer Michael, at dette kan skabe problemer, da der bliver for mange versionsnumre at holde styr på, hvis man både har versionsnummre for biblioteker og for dokumentation. Andreas argumenterer for, at dette bliver man nødt til, da der er stor forskel i, hvor ofte man udgiver nye versioner. Dette eksemplificere Andreas ved at sige, at hans JavaScript bibliotek er igang med fjerde version og spørger herefter Karsten, hvilken version hans Arduino bibliotek er på. Hertil siger Karsten, at han stadig ikke er nået til version et endnu. Først efter denne diskussion omkring versionskontrol, forklarer facilitatoren, at hensigten med versionnumret er, at den skal vise bibliotekets versionsnummeret og det derfor ikke er et udtryk for, hvilken version af dokumentationen, det er. Andreas påpeger til dette, at så skal det skrives eksplicit ved versionnummeret, hvad det henviser til. Ydermere spørger han ind til, hvad der sker hvis man først skifter version og derefter programmeringssprog. Hertil påpeger Michael, at versionsnummer og kodetype skal følges ad, når man skifter imellem dem.

Dernæst drejer samtalen sig over på *Value types*, hvor facilitatoren spørger ind til vigtigheden af denne. Her forklarer Michael, at grunden til man har lavet *Value types* til WappstoIoT er for at gøre det nemmere for brugeren at benytte biblioteket. Hertil supplere Andreas at ideen er, at man definere en type som f.eks. temperatur eller *power measurements*, således man ikke behøver at definere en basis type i koden, som f.eks. String eller number.
Karsten forklarer herefter, at disse *value types* sjældent passer til Arduino og derfor finder ham dem ikke essentielle. Karsten uddyber dette ved, at forklare at en type som lokation og farve altid vil passe til Arduino, men typer som temperatur, *voltage* og *energy* aldrig vil passe. Da deltagerne bliver spurgt ind til om dette var en problemstilling, der er unikt for Seluxit, svarer både Karsten og Andreas at det er det. Andreas forklarer, at før de lavede *value types*, definerede brugerne deres typer oftes blot som string eller Number. Dette havde den konsekvens, at Wappsto ikke vidste hvad der blev målt på. Michael

forsatte herefter med at fortælle, at problemstillingen med det er, at så kan man ikke designe brugergrænsefladen til de enkelte datatyper. Dette eksemplificere han med, at hvis Wappsto ved at det er datatypen *color* der modtages, så kan man designe en farvevælger til dette datasæt.

Efter denne forklaring retter facilitatoren fokus tilbage på forsiden og spørger igen ind til deltagernes tanker på et generelt plan. Her starter Michael ud med at forklarer at der er for mange informationer på den øverste del af forsiden. Han vil heller fortrække at flytte *Get started* sektionen op i toppen af forsiden.

Karsten siger herefter, at han mangler information om, hvad han skal bruge hvert programmeringssprog til og generelt kontekst til biblioteket. Hertil uddyber han, at han mangler noget information omkring, hvad er det man kan opnå med biblioteket og derfor bør man skifte fokus ift. hvad introduktionen forklarer nu. Hertil tilføjer Andreas, at man bør ikke først vælge programmeringssprog og så vælge hvilken case man vil arbejde med, men derimod burde man først skulle vælge case og så bagefter kan man få at vide, hvilke programmeringssprog, der er understøttet i denne case. Facilitatoren spørger herefter ind til om det ikke vil være træls, at man valgte en case og så først finder ud af sent, at den case man har valgt kun kan lade sig gøre i et programmeringssprog, som brugeren ikke har forståelse for. Hertil svarer Andreas, at det skal være andet step, man finder ud af hvilket programmeringssprog det kan udføres i.

Det næste der bliver lagt mærke til ved forsiden er, at Andreas mener at introduktionsvideoen er placeret forkert. Dertil tilføjer Karsten, at han mangler generelle informationer om brugen af biblioteket f.eks. hvordan modtager man data og sender data. Michael svarer hertil, at han ikke mener at det hører til forsiden. Hertil svarer Karsten, at de generelle ting, der ikke er unikke for programmeringssproget såsom hvordan man sender og modtager data burde blive forklaret inden eksemplerne. Karsten argumentere yderligere for dette ved, at forklarer, at kun ved at forstå det generelle brug af biblioteket, kan man forstå helt, hvordan biblioteket skal benyttes. Ydermere forklarer Andreas, at der ofte er en introduktions side i sidemenuen, hvor man vil få alle disse generelle informationer at vide. Hertil tilføjer Andreas, at det også hjælper den erfarne domæne bruger, da denne kan sorterer denne introduktion fra.

På forsiden forklarer Michael, at der bør stå meget kort, hvad det pågældende bibliotek gør og så skal der være links til en mere uddybende forklaring. Her fortæller Andreas, at man kunne erstatte de nuværende punkter under *key features* med det Michael forklarer. Andreas eksemplificere, hvordan han ser en forside bør være, ved at tilgå en dokumentation side på sin mobil. Til denne forklarer han, at der er en kort introduktion øverst på siden og derefter kommer der informationer om, hvad man kan bruge biblioteket til. Karsten tilføjer til, at man på forsiden bør give folk et hurtigt overblik over, hvad er det man som bruger får ud af at bruge dette bibliotek.

Derefter drejer samtalen sig over på den forslået introduktionsside, hvor Andreas forklarer at denne side bør være en dedikeret side til at forklare brugeren hvordan biblioteket fungerer med f.eks. datamodeller, *value types* og kommunikationen til Wappsto. Til dette forklarer Andreas, at man bør sagtens kunne komme igennem eksemplerne uden den større forståelse for den information, der er på introduktionssiden. Dog er det vigtigt, at indholdet

på forsiden fanger folks opmærksomhed, således de hurtigt får en forståelse for hvilke muligheder biblioteket giver. Karsten tilføjer til dette, at så rammer man også både de folk der bare gerne vil igang og dem der vil have en dybere forståelse for biblioteket inden de går igang.

Herefter spørger facilitatoren ind til holdningerne omkring opdelingen af *Get started* og *Quick start* siderne. Her forklarer Michael, at han ikke helt forstår opdelingen. Andreas spørger ind til om *Quick start* reelt er hurtigere end *Get started*, hvor han efterfølgende går de tre steps igennem og forklarer at han ikke mener man kan lave opsætningen hurtigere, da man skal igennem de samme steps. Michael tilføjer, at han troede at *Quick start* var til link til et eksempel, som man kan kopier direkte over i sin kode, hvilket de to andre nikker genkendende til. Karsten tilføjer til dette, at han forventende *Quick start* var til, hvis man havde oprettet konto, hentet certifikat og var klar til at påbegynde et projekt.

Ift. use cases på forsiden forklarer Karsten, at han vil gerne have byttet fokus om til, at man vælger programmeringssprog til det man gerne vil lave og ikke som det står nu, hvor man først vælger programmeringssprog og så får sine muligheder at vide derefter. Dette betvivler Michael dog og forklarer at det er nok fordi de er erfarne udviklere, at de er ligeglade med programmeringssproget. Andreas modargumenterer dog dette og siger at han heller vil have at vide, hvad er det man generelt kan og så bagefter hvilke programmeringssprog understøtter dette. Hertil forklarer Andreas, at istedet for de eksisterende *key features* på forsiden, bør punkterne hedde "Lav en enhed", "Lav en web baseret appliktation". Hertil tilføjer Michael at man også bør tilføje et punkt som hedder *High processing IoT*. På denne måde mener alle tre deltagere at man får skiftet fokus fra programmeringssprog til, hvad målet er.

Andreas og Karsten kommenterer på placeringen af, hvor man vælger programmeringssprog. Til dette forklarer de begge, at det er en god ide at man vælger programmeringssprog og så længe at knapperne følger med ned, når man scroller er det en god placering. Eneste tilføjelse til dette er, at Andreas forklarer at det bør ikke være på forsiden, men i andet step.

Dernæst scrolles der ned til *Community* sektionen. Her forklarer Karsten, at links til Github og Youtube giver god mening, dog er et link til Stack Overflow for tidligt for WappstoIoT biblioteket, da der ikke er nok brugere. Andreas tilføjer til dette, at man først går ind på Stack Overflow, når man er i problemer og det vil man ikke være, når man er på forsiden af dokumentationen. Dernæst bliver alle tre enige om, at sektionen bør blive flyttet ned i bunden af forsiden.

Til forsiden kommentere Andreas ligeledes på, at det godt at der er use cases og eksempler, så folk kan scrolle ned igennem og få sig en bedre forståelse af biblioteket end hvad den indledende introduktion har givet.

Dernæst navigere facilitatoren sig over på *Working Examples* siden (se figur 4.10) og spørger ind til tanker omkring denne. Hertil forklarer Karsten, at den bære meget præg af at være bygget på *value types* og ikke så meget *working examples*. Til dette tilføjer Andreas, at som siden er sat op nu, så vil han forvente snippets og ikke eksempler. Til spørgsmålet om hvad de vil forvente, når de kom ind på *Working Examples* siden svarer Andreas, at det vil være konkrete projekter såsom temperatur sensor, CO2 monitor og dørlås.

Dernæst navigeres der videre til *Sending Data* siden (se figur 4.20). Her argumentere Karsten for, at han vil forvente at der en kort introduktion til hvad eksemplet omhandler og kræver og så et link til kildekoden på GitHub. Andreas påpeger, at den nuværende *Sending Data* side er en step-by-step guide og ikke et klassisk working example. Efter facilitatoren gør opmærksom på, at man kan kopier koden fra hjemmesiden forklarer Karsten, at han vil forvente der mangler noget kode, hvis de kopier koden herfra. Her afbryder Michael og siger at det kan han ikke lide, hvilket Andreas giver ham ret i, men forklarer at det er sådan alle working examples er bygget op. Hertil siger Michael, at det ikke bør være en grund til at man skal gøre det på den måde, hvilket de to andre giver ham ret i.

Herefter gennemgås *Sending Data* siden, hvor alle deltagere udtaler at de godt kan lide opbygningen med, at koden er på den ene side og forklaring på den anden. Michael forslår dog, at man kunne highlighte de steder som beskrivelsen omhandler. Andreas forslår, at man kunne med fordel kunne have større afstand imellem sektionerne af koden, således det var nemmere at overskue.

Karsten tilføjer dog også, at han kun vil forvente denne type detaljegrad er gældende for et enkelt eksempel i dokumentationen. I dette eksempel skulle man så gennemgå de fleste dele af funktionaliteterne i biblioteket. Så når man kommer over i andre eksempler vil man genkende de tidligere funktionaliteter, bare brugt på en anden måde. Andreas forslår at man kan kalde dette eksempel *How to*.
Michael forslår, at man kunne lave en *How to*-guide ud fra et eksempel. Dette eksempel vil skulle tage udgangspunkt i en termostat på en radiator, hvor man kan fjernstyre varmen. Dette vil forklare, hvorfor man benytter en *control* og *report* i biblioteket, som er funktioner i WappstoIoT biblioteket. Til dette eksempel forklarer Karsten, at det er vigtigt at man forklarer, hvorfor man gør som gør og at det giver en indsigt i hvilke parametre, der kan ændres således det kan relateres til brugerens eget projekt. Ud fra denne *How-to* guide, bør der være en masse eksempler på små relevante brugsscenarier, som f.eks. hvordan man sender en tekst og vælger en farvevælger, som man som bruger så kan støtte sig op ad forklarer Michael og Karsten.

Andreas siger herefter, at han desværre er nødt til at gå og forlader lokalet. Derefter skifter samtalen over på *Value types*, hvor Karsten og Michael forklarer, at formålet med dem er at give en pæn brugergrænseflade. Michael forklarer ydermere, at de ikke er ordentlig implementeret endnu og kan derfor ikke dække alle brugsscenarier. Dernæst laves der en fælles opsummering af resultaterne, hvor facilitatoren gennemgår sin forståelse af det sagte og deltagerne giver feedback eller forklarer det sagte yderligere.

Dernæst forsættes til "I Like, I Wish, What If" metoden. Både Karsten og Michael starter med at sige, at de godt kan lide forsiden, hvis det øverste del med *key features* bliver ændret. Michael tilføjer ydermere, at han godt kan lide på *Sending data* siden, at koden og forklarende tekst er ved siden af hinanden. Der bliver dog lagt mærke til Codesandbox linket, hvor Karsten forklarer at dette ikke vil kunne lade sig gøre i dette bibliotek, da man ikke har et certifikat, så det er ikke relevant at gøre med WappstoIoT, hvilket Michael giver ham ret i. Karsten uddyber dog at på et generelt plan, er det en god ide, men ikke med dette bibliotek. Hertil forklarer Michael og Karsten, at en video er en god ide, da det henvender sig til folk, der ikke er så gode til at læse, men grundet vedligeholdelse skal eksemplerne være det rigtige til at starte med, så man ikke skal opdatere dette ofte.

Til *I wish* siger Karsten, at den primære del er konteksten øverst på forsiden, at han ønsker ændret. Dette giver Michael ham ret i. Dernæst forklarer Karsten, at et andet punkt er, at *How to* guiden burde referer til alle de ting man kan ændre ved eksemplet.

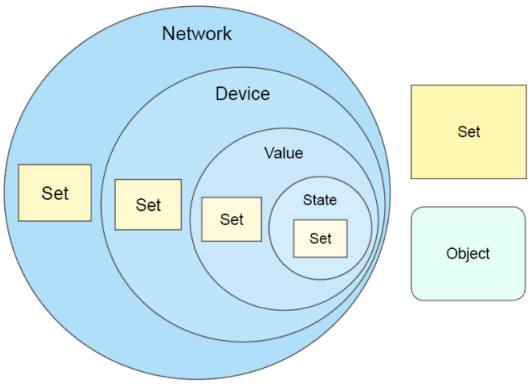
Til punktet *What if*, forklarer Michael at han ønsker at koden bliver highlightet og Karsten tilføjer at han til sidst, at han stadig mangler at få at vide, hvad hele ideen bag Wappsto er i dokumentationen.

Herefter afsluttes mødet.

Kodeeksempler E

Dette kapitel har til formål at gennemgå kodeeksemplerne, der er udviklet til *how to* guiden. Kodeeksemplerne er udviklet til Python, JavaScript og Arduino. Da alle tre kodeeksempler har overordnet set samme opbygning og samme formål, vil kun kodeeksemplet for Python blive gennemgået i kapitel. Alle tre kodeeksempler, kan findes i bilag F. Kapitlet indledes først med en forklaring af WappstoIoT bibliotekets datamodel, hvorefter kodeeksemplet for Python vil blive gennemgået.

Fælles for alle kodeeksempler er, at de tager udgangspunkt i et scenarie, hvor man som bruger ønsker at styre temperaturen via en termostat på radiatoren. I Python og JavaScript eksemplerne vil terminalen agerer termostat, hvor styringen af temperaturen vil ske via Wappsto. I Arduino eksemplet vil man skulle benytte en ESP32[1] med WiFi adgang eller en Wio Terminal[2], som skal agerer termostat. Ydermere hentes biblioteket ned lokalt ved Arduino eksemplet og der kan derved ikke benyttes de samme indbygget funktioner i biblioteket som tilfældet er med Python og JavaScript. Disse forskelle vil dog blive gennemgået i afsnit E.1.

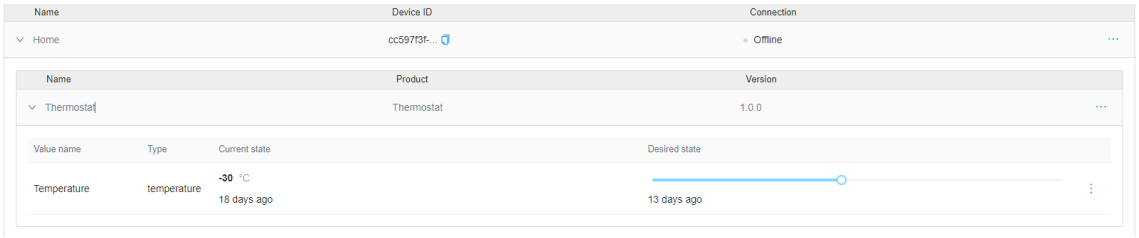


Figur E.1. Figuren viser den datamodel, der følges i udviklingen af kodeeksemplerne for WappstoIoT. Figuren er lånt fra Seluxit [2017].

På figur E.1 ses datamodelllen for biblioteket WappstoIoT. Datamodellen skal forståes på den måde, at i øverste lag af datamodellen, har man et *Network*, hvori man har flere *Devices*. Hver *Device* kan have flere *Values* og hver *Value* kan have en *State*. Et eksempel på denne datamodel er, at man har adskillige termostater i et *Network*. Hver termostat vil så være en *Device*. Hver *Device* kan have forskellige værdier, som f.eks. temperatur i

[1] `https://randomnerdtutorials.com/esp32-useful-wi-fi-functions-arduino/`
[2] `https://wiki.seeedstudio.com/Wio-Terminal-Getting-Started/`

rummet. Denne temperatur kan så have en *State*, f.eks. at den er stigende ift. den ønskede temperatur.



Figur E.2. Figuren viser et udklip af Wappsto, hvor kodeeksemplet i *how to* guiden, er udført.

På figur E.2 ses det hvorledes datamodellen kommer til udtryk for brugeren på platformen Wappsto. Ud fra dette eksempel svarer "Home" til *network* på datamodellen og "Thermostat" er *device*. *Value* er "Temperature" og *State* er de -30 grader, der står under *Current state*.

Som det ses på figur E.1, har datamodellen ydermere kasser, hvor der står "Set". Disse har til formål, at vise at man kan modificere parametre og hente tilstande i netværket. Et eksempel på dette kan f.eks. være hvis temperaturen er x og man ønsker at ændre denne til y. Her vil man så på alle niveauer kunne sætte denne temperatur, hvor valget af niveau afhænger af hvor mange enheder, man ønsker den tilstand for. "Object" er stadig under udarbejdelsen ved Seluxit og er ikke endelig defineret endnu.

Denne datamodel er obligatorisk at følge, når man benytter WappstoIoT og er derfor også inkluderet i nedenstående kodeeksempel. Hvert lag, har forskellige muligheder til indhold, men alle lag skal som minimum have et *name*.

E.1 Python

Dette afsnit har til formål, at visualisere og forklare opsætningen af WappstoIoT biblioteket for Python, samt et kodeeksemplet.

Opsætning

Opsætningen af biblioteket i Python sker ved, at brugeren først skal sørge for at de har installeret PIP. Dernæst skal de i deres fortrukne terminal kalde kommandoen "pip install -U wappstoiot". Denne kommando kan dog varierer, hvis man bruger Linux eller et andet styresystem. Dernæst skal brugeren oprette en bruger på Wappsto. Dernæst skal der hentes et certifikat i Wappsto. Når man har dette certifikat, kan man forsætte til at udføre eksempler, use cases eller andet.

Kodeeksempel

Inden kodeeksemplet kan køres, skal der først være oprettet en lokal mappe, hvori der indgår certifikater, som bliver hentet på Wappsto og en .py fil med nedenstående kode. Dernæst skal WappstoIoT installeres og terminalen, som agerer termostat, skal sættes til

at køre Python projektet, hvilket *how to* guiden har til formål at formidle. Det antages ydermere, at man har installeret PIP[3].

Den resterende del af dette afsnit vil beskrive kodeeksemplets virken via mindre snippets. Den samlede kode findes i bilag F.

```
1  import math
2  import time
3  import wappstoiot
4
5  UPDATE_INTERVAL: int = 1
6
7  temperature_reading: int = 18
8  temperature_target: int = 20
9
10 def get_temperature() -> int:
11     return temperature_reading
12
13 def set_target_temperature(target: int):
14     global temperature_target
15     temperature_target = target
16
17 def update_temperature() -> bool:
18     global temperature_reading
19     diff: int = temperature_target - temperature_reading
20     if diff:
21         temperature_reading += math.ceil(diff/3) or -1
22         return True
23     return False
```

På ovenstående snippet ses de første 23 linjer af kodeeksemplet for Python. I linje et til tre i ovenstående snippets importeres nødvendige tredjeparts biblioteker, heriblandt WappstoIoT. I linje fem initialiseres, hvor ofte den simulerede temperatur skal opdateres, når brugeren ændre på *desired state* (se figur E.2). Dernæst defineres den nuværende temperatur i linje syv og den ønskede temperatur i linje otte.

I linje 10 til 11, defineres funktionen *get_ temperature()*, som har til formål at hente den seneste temperatur indlæsning fra Wappsto via variablen *temperature_ reading()*.
Funktionen *set_ target_ temperature()* i linje 13 - 15, har til formål at definere variablen *target* ud fra, hvad *desired state* i Wappsto er sat til.

Funktionen *Update_ temperature* i linje 17 - 23 har til formål at justere variablen *temperature_ reading*, hvis der er forskel på *temperature_ reading* og *temperature_ target*. Ydermere returnere funktionen en boolean om forskellen er sand eller falsk, som bruges i *def main():* til at definere om der skal sendes data til Wappsto.

Denne del er ens for alle tre kodeeksempler. Nedenstående snippet viser de næste 39 linjer af koden.

```
1  def createThermostat() -> wappstoiot.Value:
2
3      network = wappstoiot.createNetwork(
4          name="Home",
5          description="Thermostat Simulator"
6      )
7
8      thermostat_device = network.createDevice(
```

[3] https://pypi.org/project/pip/

```
9         name="Thermostat",
10        description='Thermostat Simulator',
11        product='Thermostat',
12        version='1.0.0',
13        manufacturer='ME',
14    )
15
16    temperature_value = thermostat_device.createValue(
17        name="Temperature",
18        permission=wappstoiot.PermissionType.READWRITE,
19        value_template=wappstoiot.ValueTemplate.TEMPERATURE_CELSIUS
20    )
21
22    temperature_value.report(
23        get_temperature()
24    )
25
26    temp = temperature_value.getControlData()
27    if isinstance(temp, float):
28        set_target_temperature(int(temp))
29
30    @temperature_value.onControl
31    def control_temp(obj: wappstoiot.Value, data: Union[float, str, None]):
32        if isinstance(data, float):
33            set_target_temperature(int(data))
34
35    @temperature_value.onRefresh
36    def refresh_temp(obj: wappstoiot.Value):
37        obj.report(get_temperature())
38
39    return temperature_value
```

I linje 1 på ovenstående snippet defineres funktionen *createThermostat*. Denne funktion, er hvor datamodellen på figur E.1 bliver fulgt. Her defineres der først et *Network* i linje 3 - 7, som blandt andet bliver navngivet "Home".

I linje 8 - 14, defineres *Device*, hvori enheden bliver navngivet og bliver beriget med blandt andet en beskrivelse. I linje 16 - 20, defineres *Value*, som bliver navngivet "Temperature". Hvorledes alle disse informationer bliver præsenteret til brugeren kan ses på figur E.2. Denne del er anderledes for JavaScript og udførelsen for dette sprog, kan ses på figur E.3.

I linje 22 - 24, sendes den nyeste temperatur måling til Wappsto via den indbygget funktion i WappstoIoT "value.report()".

I linje 26, initialiseres variablen "temp" først til den indbygget funktion i WappstoIoT, som hedder ".getControlData". Denne funktion har til formål at hente den seneste kontrolværdi fra Wappsto. Dernæst er der lavet et *if* statement i linje 27 - 28, der ændre "target_temperature" værdien og typen til en *integer*, hvis den modtaget værdi er en *float*.

I linje 30 - 33 sætte en callback funktion kaldet "control_temp". Denne funktion har til formål, at ændre "target_temperature" værdien i tilfælde af ændringer modtaget fra Wappsto. ".onControl" er ligeledes en indbygget funktion i WappstoIoT biblioteket.

Hvis der trykkes på knappen *Refresh* på brugergrænsefladen i Wappsto, kaldes funktionen i linje 35 - 37. Denne funktion har til formål, at opdatere den nuværende temperatur værdi i Wappsto. ".onRefresh" er ligeledes en indbygget funktion i WappstoIoT biblioteket.

Til sidst i funktionen returnere værdien for variablen "temperature_value". Denne del af

koden er ens for JavaScript, hvor Arduino eksemplet er opbygget anderledes. Dette bliver gennemgået i afsnit E.3.

Nedenstående snippet er den sidste del af koden for kodeeksemplet for Python.

```
1  def main():
2      wappstoiot.config(
3          config_folder="config"
4      )
5      temperature_value = createThermostat()
6
7      try:
8          while True:
9              time.sleep(UPDATE_INTERVAL)
10             if update_temperature():
11                 temperature_value.report(
12                     get_temperature()
13                 )
14     except KeyboardInterrupt:
15         pass
16     finally:
17         wappstoiot.close()
18
19 if __name__ == "__main__":
20     main()
```

I ovenstående snippet kaldes "main()", hvori der først hentes certifikater i linje 2 - 4. Dernæst kaldes en *try block*, hvori der er et *while loop*. Hvis dette *while loop* er sandt, så sættes systemet til at vente i tiden defineret i variablen "UPDATE_INTERVAL". Dernæst hvis *if statementet* er sandt rapporteres den seneste værdi til Wappsto.

Denne del er ens for alle tre programmeringssprog.

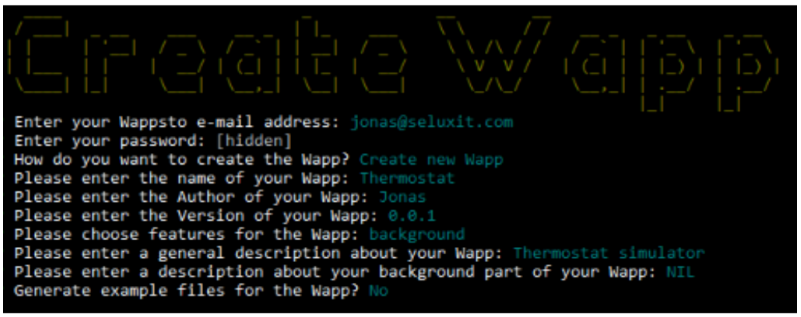
E.2 JavaScript

Dette afsnit har til formål, at forklare opsætningen af WappstoIoT biblioteket for JavaScript, samt en gennemgang af de nødvendige steps i *How to guide*.

Når JavaScript projektet på *How guide* siden skal installeret, skal man som bruger have installeret node.js[4].
Dernæst skal brugeren oprette en Wappsto konto. I terminalen, skal brugeren installere både en *wapp creator* og et bibliotek der håndtere certifikater. Dette gøres via følgende kommandoer; "npm i –save wappsto-wapp" og "npm i -g wappsto-cli". Når disse to biblioteker er installeret er man som bruger klar til at lave sin egen webbaseret applikation.

Her skal brugeren tilgå terminalen og indtaste følgende kommando; "npx wapp create".

[4] `https://nodejs.org/en/`



Figur E.3. Figuren viser de informationer som brugeren skal indtaste i terminalen for at lave en webbaseret applikation. På figuren er der givet et eksempel på, hvordan disse skal udfyldes, hvilket er det der er skrevet med grønt.

Fra denne kommando, skal brugeren indtaste de informationer som ses på figur E.3. Herfra skal brugeren kopier kodeeksemplet over i en main.js fil, som er placeret i projektmappen under "Background". Det sidste brugeren skal gøre for at får projektet til at køre er at indtaste følgende kommando, imens man er navigeret til projektetmappen; "npx wapp serve".

E.3 Arduino

Dette afsnit har til formål, at forklare opsætningen af WappstoIoT biblioteket for Arduino, samt en gennemgang af de nødvendige steps i *How to guide*.

En forudsætning for at man kan installere opsætte WappstoIoT i Arduino er, at man har installeret arduinojson[5]. Det vil blive antaget at man som bruger har installeret det nødvendige hardware på forhånd.
Dernæst skal brugeren oprette en Wappsto konto og deri hente certifikater til projektet i Wapp'en "IoT Certificate Manager". Dernæst skal selve wappsto biblioteket hentes på GitHub[6] og inkluderes i Arduino. Dernæst skal kodeeksemplet fra *How to Guide* kopiers over i projektet og man kan derefter køre projektet.

Arduino er i visse dele af kodeeksemplet opbygget unikt ift. Python og JavaScript eksemplerne. De mest væsentlige forskelle vil blive gennemgået via nedenstående snippet.

```
1  DeviceDescription_t thermostatParameters = {
2      .name = "Thermostat",
3      .product = "Thermostat",
4      .manufacturer = "ME",
5      .description = "Thermostat Simulator",
6      .version = "1.0.0",
7      .serial = "",
8      .protocol = "",
9      .communication = "",
10 };
11
12 ValueNumber_t temperatureParameters = {
13     .name = "Temperature",
14     .type = "temperature",
15     .permission = READ_WRITE,
16     .min = -30,
```

[5] https://arduinojson.org/
[6] https://github.com/Wappsto/arduino_wappstoiot

```
17      .max = 50,
18      .step = 1,
19      .unit = "C",
20      .si_conversion = "[K] = [C] + 273.15"
21  };
```

Da koden skal placeret lokalt, kan flere prædefineret funktioner ikke benyttes i Arduino eksemplet. Ovenstående snippet viser de parametre, som er blevet defineret for "Device" og "Value". I disse parametre indgår blandt andet temperatur i celsius og en konvertering fra celsius til fahrenheit, samt hvilket interval temperaturen må være i.

Disse informationer er prædefineret i både Python og JavaScript.

```
1   void initializeWifi(void)
2   {
3       Serial.println("Initializing WiFi");
4       WiFiMulti.addAP(ssid, password);
5       while(WiFiMulti.run() != WL_CONNECTED) {
6           Serial.print(".");
7           delay(500);
8       }
9       Serial.print("IP address: ");
10      Serial.println(WiFi.localIP());
11  }
```

Ydermere skal der initialiseres til et WiFi netværk, som bliver gjort via ovenstående funktion.

Usability test F

Det kapitel har til formål, at beskrive demografiske oplysninger om de forsøgspersoner som er blevet brugt i usability forsøgene beskrevet i kapitel 5. Ydermere har kapitlet til formål at dokumentere introduktionen til begge delopgaver udført i usability forsøgene.

F.1 Information om forsøgspersoner

Følgende afsnit har til formål at give indsigt i forsøgspersonernes demografi og erfaring indenfor programmering. Senest 24 timer før forsøgene skulle udføres modtog forsøgspersonerne et spørgeskema via et link, der omhandlede demografiske spørgsmål, samt spørgsmål om forsøgspersonens egen vurdering af erfaringsniveau jf. Nielsen [1993][s. 43 - 47] tre områder af ekspertise (se figur 2.1). Spørgeskemaet er udarbejdet i SurveyXact[1] og indeholdte ni spørgsmål i alt.
Spørgeskemaet kan findes i bilag G.

FP Nr.	Iteration	Aldersgruppe	Type af udvikler	Erfaring (År)	Primære programmringssprog
PFP1	Pilot	Over 49	I used to be a developer by profession, but no longer am	10+	C and C#
FP1	1	20-39	I am a student who is learning to code	3-5	Java, JavaScript, HTML & CSS
FP2	1	20-39	I am a student who is learning to code	0-2	JavaScript
FP3	2	20-39	I am a student who is learning to code	3-5	JS, HTML, CSS
FP4	2	20-39	I am a student who is learning to code	6-9	CPP, C#,Python, Arduino
FP5	3	20-39	I am a student who is learning to code	6-9	C++, Python, Matlab, Matlab, C#, Arduino
FP6	3	20-39	I am not primarily a developer, but I write code sometimes as part of my work	0-2	Excel VBA, Pythoon
FP7	3	20-39	I am a student who is learning to code	3-5	CPP, Python, DAX
FP8	3	20-39	I am not primarily a developer, but I write code sometimes as part of my work	3-5	Python, Arduino and C++

Tabel F.1. Tabellen viser hvilke iterationer forsøgspersonerne har deltaget i, samt deres aldersgruppe, type af udvikler, års erfaring og deres primære programmeringssprog. PFP1 har forsøgspersonen, der er brugt til pilotforsøg.

Som det ses i tabel F.1, er alle forsøgspersoner, der har deltaget i iterationerne, været imellem 20 og 39 år. Yderemere ses det, at alle forsøgspersoner vil jf. definitionen i afsnit 3.2, være en del af gruppen "Deltidsuviklere". Ud fra deres eget udsagn er der fem forsøgspersoner, der blandt andet ser Python, som deres primære programmeringssprog og tre forsøgspersoner, der ser Javascript som deres fortrukne programmeringssprog.

[1] `https://www.survey-xact.dk/`

FP Nr.	Domæne	Programmering	Brug af computer
PFP1	30	42	100
FP1	0	40	98
FP2	0	20	40
FP3	4	80	95
FP4	0	46	90
FP5	0	50	90
FP6	0	21	100
FP7	1	54	100
FP8	0	50	80

Tabel F.2. Tabellen viser forsøgspersoners egen vurdering af deres ekspertise niveau indenfor de tre områder som Nielsen [1993][s. 43 - 47] fremhæver. Her er det gældende for alle tre områder, at 0 betyder slet ikke erfaren og 100 betyder meget erfaren.

Det ses i tabel F.2, at alle forsøgspersoner, der er brugt i iterationerne, er uerfarne indenfor Wappsto domænet. Yderemere ses det, at FP2 er den forsøgsperson, der ser sig selv som den mest uerfarne indenfor programmering, da denne kun har vurderet sig selv til 20. Den der har vurderet sig selv højest er FP3, som har givet sig selv 80. Generelt indenfor brug af computer har FP2 igen vurderet sig lavest med 40, hvor både FP6 og FP7 har vurderet sig selv til 100.

F.2 Introduktion

Dette afsnit er delt op i to dele. hvor der først vil blive gennemgået introduktionen til delopgave 1.
Dernæst vil introduktionen til delopgave 2 blive gennemgået.
Fælles for begge, er at introduktionen ikke blev fulgt ordret, men alle dele af introduktionen blev forklaret. Ydermere blev forsøgspersonerne forklaret, at der tages udgangspunkt i en prototype og derfor vil dele af prototypen ikke virke og dette vil blive vist med en "Working in progress" side.

Delopgave 1
"Det du skal forstille dig nu her er, at du sidder derhjemme og gerne vil lave et hobbyprojekt, der omhandler et smart home.
Du har ledt efter nogle biblioteker, du kan bruge til dette projekt og har fundet det her, som du gerne vil bruge. Nu vil du egentlig bare gerne igang med at bruge biblioteket, så nu skal du bare gå igang med at opsætte biblioteket, hvilket er din første opgave."

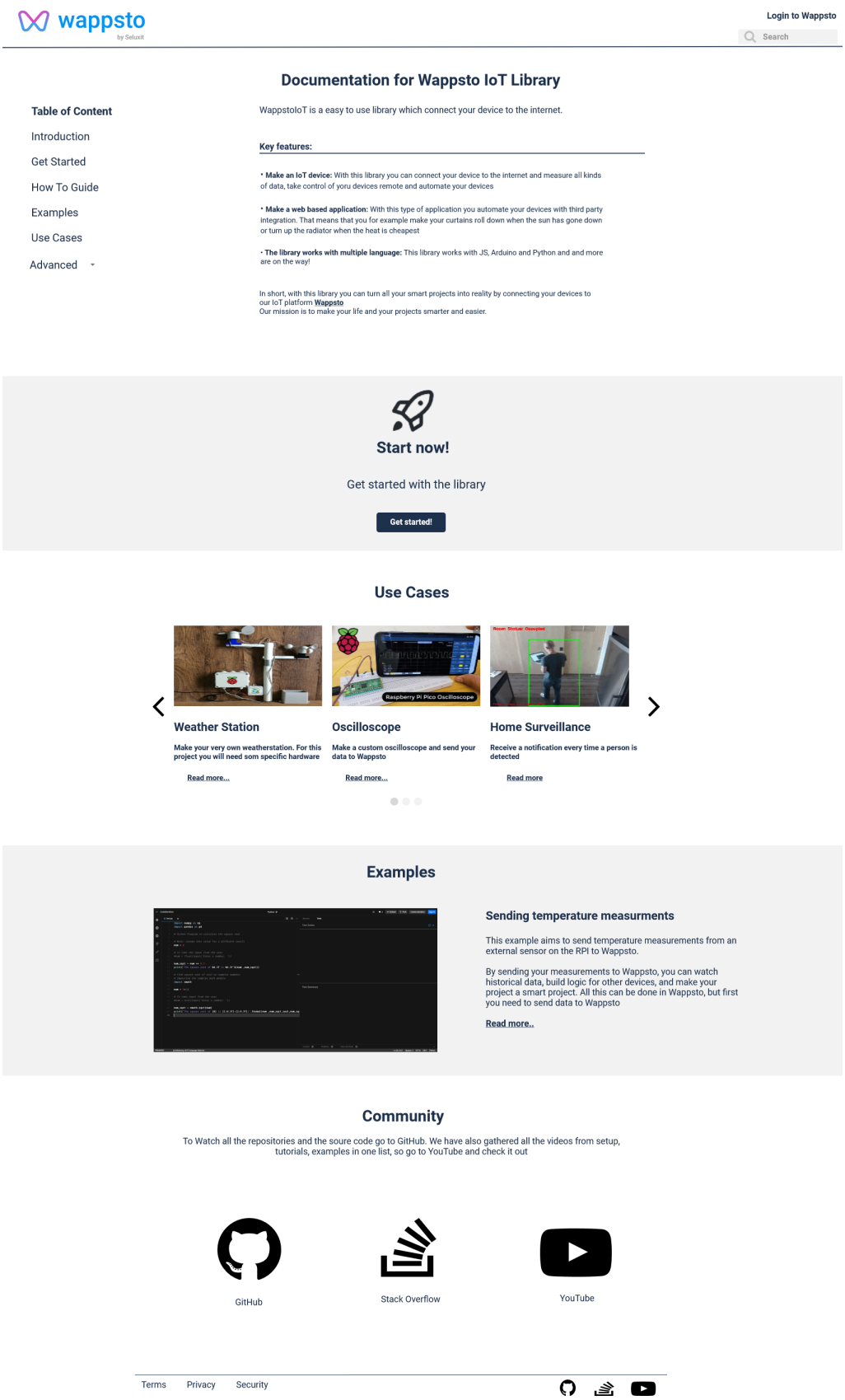
Delopgave 2
"Nu har du fået installeret biblioteket og nu vil du egentlig bare igang. Du vil have opsat noget og se det virke."

5 version - Gennemgang af strukturen

Dette kapitel har til formål at gennemgå den seneste version af prototypen for dokumentation for tredjeparts biblioteker. Gennemgangen vil have fokus på strukturen og hvilke fokuspunkter, der har været på de pågældende sider. Dette er grundet projektet overordnet formål med at designe et generisk design, der kan bruges af lignede biblioteker.

Kapitlet vil gennemgå alle prototypens sider, hvor flowet er holdt den samme, som vist på figur 5.4. Link til prototypen kan findes i bilag D i .txt filen "Links til prototyper".

Fælles for alle sider er, at det er forsøgt at aktivere brugeren med hands-on erfaring, så tidligt som muligt i dokumentationen, samt at dokumentationen kun beskriver de nødvendig informationerne. Yderemere har der været særlig fokus på at forklare formål med biblioteket og de handlinger der skal udføres for at benytte det. Ydermere er det grundet teorien gennemgået i afsnit 2.3 tilføjet flere interaktionsmuligheder for brugerne. Dette har til formål, at tilgodese forskellige ekspertiseniveauer.



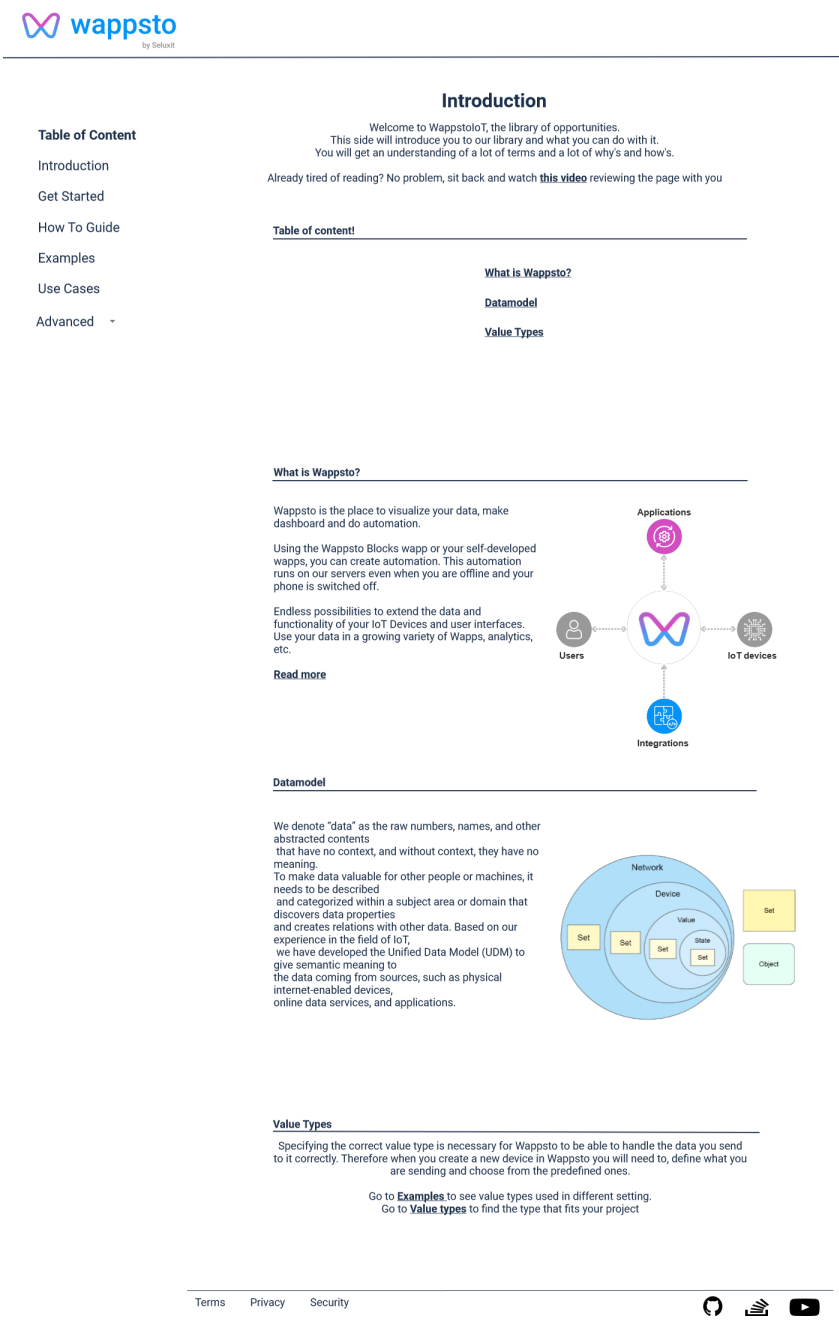
Figur G.1. Figurgen viser forsiden.

På figur G.1 ses det, at øverst på forsiden af prototypen er det valgt at beskrive nøgle

funktioner, som biblioteket tilbyder. Dette er valgt ud fra den feedback som blev modtaget af udviklerne i afsnit 4.5.1. Her fortalte udviklerne, at der skulle være særlig fokus på, at man som bruger hurtigt kunne finde ud af om det er relevant for brugeren.

Under forklaringen af biblioteket, kan brugeren tilgå opsætningsguiden i sektionen med *Get started* knappen. Denne placering er valgt ud fra både den heuristiske evaluering, brugerundersøgelser og feedback fra udviklere, der alle forklarede at en nem setup er højt prioriteret og man burde kunne tilgå opsætningen nemt.

Dernæst vises use cases og eksempler, som begge har til formål, at give et yderligere indsigt i formål og brug af biblioteket. Til sidst på siden vises links til GitHub, Stack Overflow og YouTube, som alle har til formål at give brugeren yderligere indblik i biblioteket, samt at hjælpe brugeren til at finde steder, hvor man kan fejlsøge.



Figur G.2. Figuren viser *Introduction* siden.

Som det ses på figur G.2 er der udarbejdet en introduktions side, som har til formål, at give brugeren et mere avanceret indblik i bibliotekets brug. Denne side er henvendt til den erfarne bruger, som både ønsker en overordnet forståelse for hvad Wappsto er og kan, men også dem der ønsker at undersøge datamodellen og *value types*. En gennemgang af datamodellen kan findes i appendiks E, Til denne side, er der både designet for forklarende tekst tilhørende illustrationer, samt en video, hvor hensigten er at gennemgå alle punkter på siden. Denne video er dog ikke udarbejdet endnu, men er blot designet for øverst på siden, hvor den kan tilgås via et hyperlink.

G.1 Python

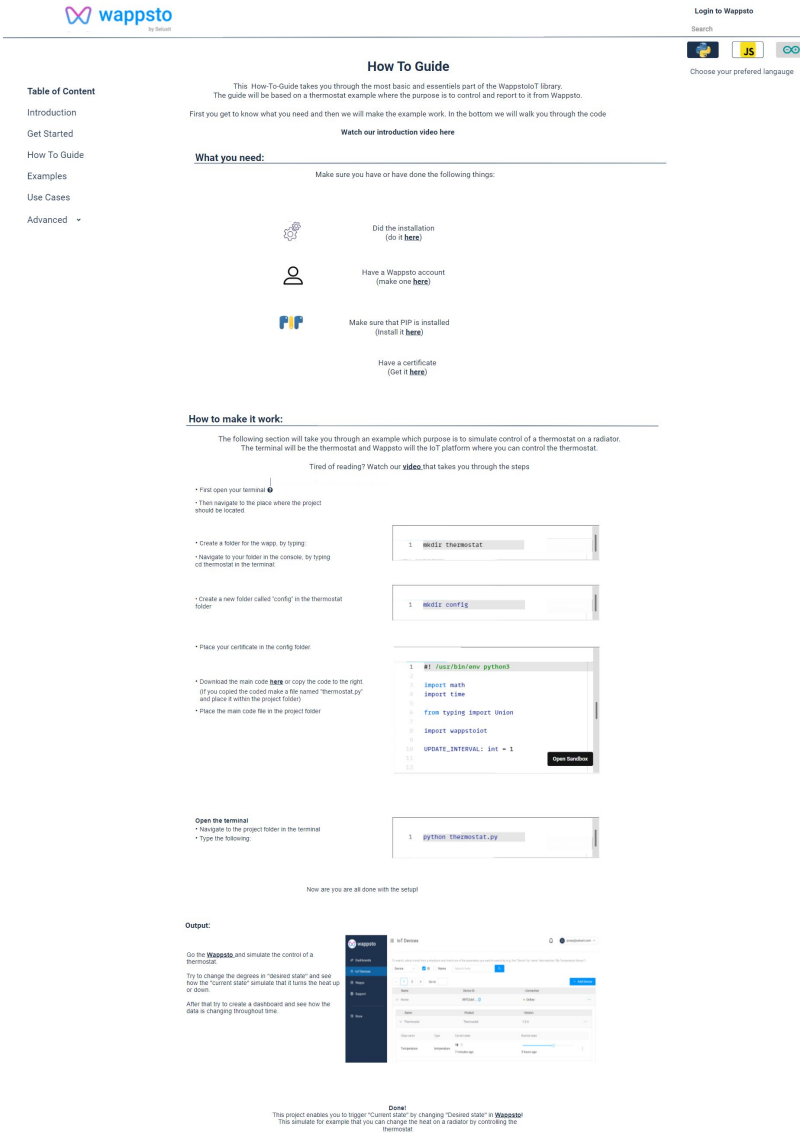
Følgende afsnit vil gennemgå både opsætning og *how to* guiden for programmeringssproget Python.

Figur G.3. Figuren viser opsætningsguiden til Python.

Når man trykker på enten *Get Started* knappen eller i indholdsfortegnelsen på forsiden vil man blive ført til opsætningsguiden vist på figur G.3. Det ses på figur G.3, at opsætningguiden til Python er opdelt i fire steps. Hvert step har fået deres egen overskrift for at give den mere erfarne bruger mulighed for at navigere i dokumentationen efter netop det denne søger. Ydermere er opdelingen udarbejdet således at i første step, vælges der programmeringssprog. I step 2 oprettes der en Wappsto konto, hvorefter biblioteket skal installeres i Python i step 3. I step 4 skal hentes et certifikat ned, som skal bruges til ens projekt. Her bliver steppet indledningsvis startet med en forklaring på, hvorfor dette skal hentes og dernæst er steppet opdelt i to. Venstre side er alle steps skrevet ned i punktform og i højre side, er der en video, der viser hvorledes certifikaterne skal hentes. Her er det igen hensigten, at den mere erfarne bruger kan skimte sig igennem stepsene og den mere

uerfarne kan blive guidet via videoen.

Til sidst på siden, henvises der til hvad det næste naturlige skridt kunne være efter opsætningen. Her bliver der via ikoner og tekst linket til indtroduktionssiden og *how to* guiden. Dette har til formål, at give brugeren mulighed for at få inspiration til, hvad næste step kunne være.



Figur G.4. Figuren viser øverste halvdel af *how to* guiden til Python.

How to guiden, som ses på figur G.4, indledes med en introduktion med formålet for denne guide. Ligeledes har brugeren en mulighed for at trykke på et hyperlink og få formålet præsenteret via en video. Dernæst gennemgås, hvilke forudsætninger der kræves for, at gennemføre denne guide. Brugeren får forudsætninger præsenteret både via ikoner og tekst og der er links til de steder, hvor man kan udføre punkterne.

For at tage forbehold for *active user paradox*, er det valgt at man først får et eksempel til at virke, for derefter at få introduktion til hvordan det virker. Her indledes opsætningen af projektet i punkt "How to make it work", hvor brugeren både har muligheden for at følge step by step guiden eller se videoen, hvor alle handlinger bliver gennemgået. Til sidst i dette step, kan brugeren se det forventet resultat af forsøget.

Selvom figur G.4 og G.5 vises på to forskellige figurer, er de i prototypen en enkelt side, hvor figur G.5 er sat i forlængelse af G.4.

Figur G.5. Figuren viser nederste halvdel af *how to* guiden til Python.

I steppet "The code walkthrough", som ses på figur G.5 gennemgås, hvordan koden virker. Her kan brugeren igen få informationerne om projektets virken via en video eller ved at gennemlæse den komplette kode, hvor der er kommentarer i venstre side.

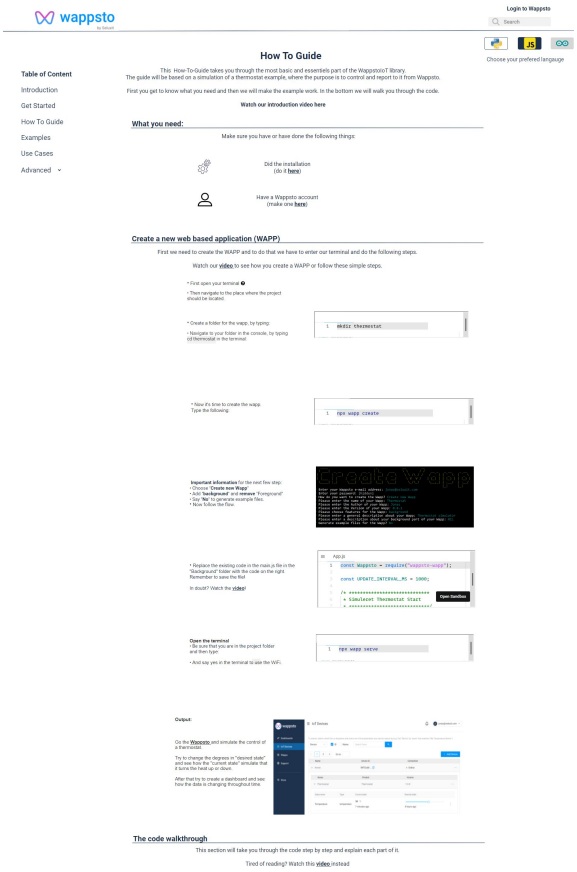
Til sidst på siden får brugeren igen anbefalinger om næste step, som har til formål at skabe konsistens i designet og et naturligt flow for brugen af biblioteket. På denne side refereres der til datamodellen og eksempler.

G.2 JavaScript

Følgende afsnit vil gennemgå både opsætning og *how to* guiden for programmeringssproget JavaScript.

Figur G.6. Histogrammet viser opsætningsguiden til JavaScript.

Hvis brugeren trykker på JavaScript ikonet i step 1 i opsætningsguiden, vil de blive navigeret til siden vist på figur G.6. Denne side har samme opbygning og struktur som beskrevet for opsætningsguiden til Python, som er gennemgået i afsnit G.6. Dette har igen til formål, at skabe konsistens i designet, hvor de samme ikoner benyttes, samt opdelingen af siden, som tilfældet er for Python. Eneste forskel er antallet af steps, da der ikke skal hentes et certifikat i JavaScript, som tilfældet er med Python.



Figur G.7. Figuren viser den øverste halvdel af *how to* guiden til JavaScript.

How to guiden for JavaScript, som er vist på figur G.7 og G.8, har igen samme opbygning som *how to* guiden for Python. Dette er igen valgt for at skabe konsistens i designet. Eneste forskelle er selve handlingerne der skal udføres ellers er *how to* guiden bygget strukturel op på samme måde som *how to* guiden til Python.

Figur G.8. Figuren viser den nederste halvdel af *how to* guiden til JavaScript.

Den nederste del af *how to* guiden, som ses på figur G.8 bliver gennemgået på samme måde som *how to* guiden for Python. Eneste forskel er selve koden.

Bilagoversigt H

Følgende kapitel indeholder en oversigt over projektets vedhæftede bilag.

Bilag A: Rådata fra brugerundersøgelser
Bilag B: Design studio workshop
Bilag C: Heuristisk Evaluering
Bilag D: Prototype versioner
Bilag E: Feedback fra Udviklere
Bilag F: Kode
Bilag G: Usability test

Alle bilag har undermapper, der yderligere opdeler informationerne.