

Kandidatafhandling

Jeppe Poss

Open source: Kritiske valg

Copenhagen Business School

Cand.merc.(it) IT Management and Business Economics

Vejleder: Mogens Kühn Pedersen

Afleveringsfrist: 22. juni 2012

English title: Critical choices in open source software

Forfattererklæring

(indsættes som side 2 i opgavebesvarelsen)

Identifikation	
Opgavens titel:	Open source: Kritiske valg
Studium:	Cand.merc.it (IMBE)
Fag:	Kandidatafhandling
Prøveform (fx Hjemmeopgave eller bachelorprojekt):	
Kandidatafhandling	

CPR nr.:	Navn på forfatter(e):	Evt. gruppenr.
	Jeppe Poss	

Opgavens omfang:
Efter eksamensbestemmelserne for prøven må opgavebesvarelsen højst bestå af 80 sider [sideantallet indsættes] eksklusiv forside, litteraturliste og evt. bilag. Bilagene indgår ikke i bedømmelsen. Opgavens sider må i gennemsnit ikke indeholde mere end 2.275 anslag (inkl. mellemrum). (Fx svarende til 35 linjer à 65 anslag). Siderne skal have en margen på mindst 3 cm. i top og bund og mindst 2 cm. i hver side. Skrifttypen skal være på minimum 11 pitch. Tabeller, figurer, illustrationer og lign. tæller ikke med i antallet af anslag, men berettiger ikke til at overskride det fastsatte maksimale sidetal.

Code of conduct:	
Undertegnede forfatter(e) erklærer herved: - at jeg/vi selvstændigt eller i samarbejde med ovenfor anførte gruppemedlemmer har udført arbejdet med denne opgave og skrevet den indeholdte tekst - at jeg/vi har markeret alle citater med citationstegn og forsynet dem med referencer til deres oprindelse - at opgaven overholder de ovenfor anførte vilkår for opgavens omfang og udformning.	
Dato: 22 juni 2012	Underskrift(er):

Critical choices in open source software

By Jeppe Poss

Master thesis

Abstract

In this thesis the main goal was to identify certain choices that would be characterised as critical for open source software (OSS) projects. The thesis used motivation, organisation, and the technological growth of the project as the means to determine whether a choice would be critical for a given project. As a benchmark the common-based peer-production process (Andreev, Feller, Finnegan, & Moretz, 2010) was used. The findings revealed seven generic decision categories; license, architectural (software), distributions, management, cultural, external environment, and when to stop the project.

The three projects that were included in this thesis were all selected for the reason that they would be able to provide a different point of view on OSS development.

A key finding was how much of an impact the culture of the projects had on how well it performed. Two of the selected projects, both end user programs, began in a commercial context and both were struggling with a culture that made it hard for others to be a part of the community. The third project, a Linux distribution, used culture as a mean to control the project, which turned out to work great. There was however some undefined factors that could help explain why two projects were struggling with getting people involved as developers. One of which would be to define the type of software that the project was developing.

The findings should further the understanding of how critical choices are constituted and help both researchers and practitioners better understand the dynamics of OSS development.

Keywords: Open source software, critical choices, commons-based peer-production, motivation, organisation, culture, technological growth.

Open source: Kritiske valg

Indholdsfortegnelse

1	Introduktion	10
2	Begrebsafklaring	10
2.1	Open source software	10
2.2	Community	11
3	Problemformulering.....	12
4	Emneafgrænsning	14
4.1	Aktive community.....	14
4.2	Komplementære teknologier	14
4.3	Internet.....	14
4.4	Internationalisering (i18n).....	15
4.5	Open source historiske bølge	15
4.6	Omgivelser	15
4.7	Kritiske valg.....	15
5	Videnskabeligmetode	16
5.1	Paradigme.....	16
5.1.1	Epistemologi	17
5.1.2	Ontologi	17
5.1.3	Metodologi	18
5.2	Forskningsprocessen	18

6	Den videnskabeligmetodes betydning for undersøgelsen	19
6.1	Formålet med forskningen	19
6.2	Lokalisering og intuitiv karakteristik af fænomenet.....	19
6.3	Empiriske erfaringer af fænomenet	20
6.4	Abduktion af de konstituerende mekanismer.....	20
6.5	Præsentation af den hypotetiske konklusion vedrørende konstituering af fænomenet.....	20
7	Teori.....	21
7.1	CBPP processen	21
7.2	Motivation	23
7.2.1	Motivationsudfordringen	23
7.2.2	Indre motivation	24
7.2.3	Ydre motivation	25
7.3	Organisation	26
7.3.1	Organisationskultur	26
7.3.2	Organisationsstruktur.....	29
8	OSS projektanalyse rammeværk.....	30
8.1	Organisationsstruktur.....	30
8.2	Kultur	30
8.3	CBPP processen	31
8.4	Kritiske valg.....	31
9	Empiri.....	32
9.1	Valg af projekter	32
9.1.1	Bromine	33
9.1.2	Varnish Cache	33
9.1.3	Exherbo.....	34

9.2	Interviewguide.....	34
9.3	Interviewoversigt.....	36
9.4	Sekundære kilder.....	36
9.5	Validitet	37
9.6	Reliabilitet.....	37
10	Bromine	37
10.1	Organisationsstruktur.....	38
10.1.1	Eniro 2007-2010	38
10.1.2	Bromine 2010-2011	39
10.1.3	Arbejdsdeling.....	39
10.2	Kultur	40
10.2.1	Artefakter	41
10.2.2	Værdier	44
10.2.3	Grundlæggende antagelser	46
10.2.4	Det kulturelle paradigme.....	47
10.3	CBPP.....	48
10.4	Kritiske valg.....	50
10.4.1	K1 - Licensvalg.....	50
10.4.2	K2 + K6 - Arkitektur.....	50
10.4.3	K3 + K7 - Autonom ledelse.....	51
10.4.4	K4 - OpenQA	52
10.4.5	K5 + K8 - Distributioner.....	52
10.4.6	K9 - Projektet stoppes	52
10.4.7	Kritiske valg sammenfattet.....	53

11	Varnish Cache	53
11.1	Organisation	54
11.1.1	Arbejdsdeling	56
11.2	Kultur	57
11.2.1	Artefakter	57
11.2.2	Værdier	60
11.2.3	Grundlæggende antagelser	60
11.2.4	Det kulturelle paradigme.....	61
11.3	CBPP processen	62
11.4	Kritiske valg.....	64
11.4.1	K1+ K7 - Licensvalg.....	64
11.4.2	K2 - Kultur - Vi skal være de bedste.....	64
11.4.3	K3 - Distributioner	65
11.4.4	K4 - Kode hygiejne	65
11.4.5	K5 - VUG.....	66
11.4.6	K6 - VMODs.....	66
11.4.7	K8 - Bestyrelse	67
11.4.8	Kritiske valg sammenfattet.....	67
12	Exherbo	68
12.1	Organisation	68
12.1.1	Arbejdsdeling.....	69
12.2	Kultur	70
12.2.1	Artefakter	70
12.2.2	Værdier	72
12.2.3	Grundlæggende antagelser	73
12.2.4	Det kulturelle paradigme.....	74

12.3	CBPP processen	75
12.4	Kritiske valg.....	77
12.4.1	K1 - Kulturelle værdier.....	77
12.4.2	K2 - Overgang til git	77
12.4.3	K3 - Manifest.....	78
12.4.4	K4 - Forventninger	78
12.4.5	Kritiske valg sammenfattet.....	79
13	Konfigurationer	79
14	Generalisering af kritiske valg.....	81
14.1	Licensvalg.....	81
14.2	Arkitektur.....	82
14.3	Distributioner	83
14.4	Ledelse.....	83
14.5	Kultur	84
14.6	Omverdenen.....	84
14.7	Stop projektet.....	85
15	Konklusion.....	85
16	Diskussion	88
17	Referencer.....	89
18	Bilag	92
Bilag 1.	SourceForge.net	92
Bilag 2.	Apache.org.....	92
Bilag 3.	Wikipedia - Varnish cache	92
Bilag 4.	Varnish Software - Social media	92

Bilag 5.	Varnish Software - Online newspapers	92
Bilag 6.	Bromine README fil	93
Bilag 7.	VMods.....	93
Bilag 8.	VUG1.....	93
Bilag 9.	VMods liste	94
Bilag 10.	Rettigheder i forhold til GPL	94
Bilag 11.	Exherbo fysiske udtryk: Hjemmeside	95
Bilag 12.	Exherbo repositories.....	95
Bilag 13.	Exherbo Social Engineering	96
Bilag 14.	Exherbo Manifest	97
Bilag 15.	Interview guide.....	98
Bilag 16.	Mind map til interviewguide.....	100
Bilag 17.	Bromine patch opfordring	101
Bilag 18.	Release candidate.....	101
Bilag 19.	Varnish Forum	101
Bilag 20.	Varnish VUG1.....	103
Bilag 21.	Bromine kritiske valg oversigt	103
Bilag 22.	Varnish kritiske valg oversigt.....	105
Bilag 23.	Exherbo kritiske valg oversigt.....	106
Bilag 24.	Interview.....	107
Bilag 25.	Varnish antal bidrag.....	107

1 Introduktion

Inspirationen til denne afhandling kom i forbindelse med forfatterens tidligere involvering i et open source software (OSS) projekt. I dette projekt blev der taget en række valg, som fik konsekvenser for projektet. I sidste instans betød det at projektet stod overfor valget om projektets skulle fortsætte eller stoppe.

Hensigten med denne afhandling var derfor at undersøge hvilke kritiske valg der opstod i OSS projekter, og ud fra disse valg forsøge at generalisere dem, så andre OSS iværksættere kunne drage fordel af disse erfaringer.

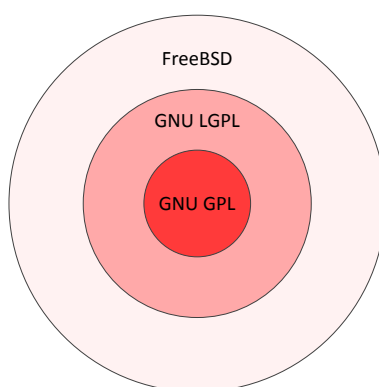
For at skabe et mere alsidigt billede fandt forfatteren det nødvendigt at inddrage en række andre projekter, som på hver deres måde kunne bidrage til en bedre forståelse af hvilke kritiske valg der opstår i OSS projekter.

2 Begrebsafklaring

I dette afsnit er der redegjort for de væsentlige begreber, som er præsenteret i denne afhandling.

2.1 Open source software

Når open source software (OSS) skal defineres, er det et spørgsmål om licensvalg. Indenfor OSS skelnes der imellem to typer licenser, henholdsvis restriktive og permissive licenser. Der findes desuden licenser, der forsøger at lave et kompromis imellem disse ekstremer. De kaldes hybridlicenser. Figur 1 skitserer, hvor meget frihed man har, hvis man vælger en given licens. Der findes en lang række licenser. Her er taget udgangspunkt i tre forskellige, henholdsvis FreeBSD, GNU LGPL og GNU GPL.



Figur 1 Graden af frihed

FreeBSD licensen¹ er en permissiv licens, hvilket vil sige, at man må bruge softwaren til hvilket som helst formål, så længe man overholder retningslinjerne for licensen. Denne type licens har de fleste anvendelsesmuligheder for andre, da man kan udgive sit eget program, som benytter software udgivet under FreeBSD licensen under en selvvalgt licens.

GNU GPL² er en restriktiv licens. Dette betyder, at man er bundet af en række rettigheder, som man skal videregive, hvis man vælger at benytte software udgivet under denne licens. Denne type licens har de færreste anvendelsesmuligheder for andre. Hvis man fx vil inkorporere et GPL program i sit eget, skal ens eget program udgives under samme GPL licens.

GNU LGPL³ kan betegnes som en hybrid licens. Den adskiller sig fra de to andre typer ved at være bundet af samme licensbetingelser som GNU GPL men med en undtagelse, som gør, at man må inkorporere softwaren i sit eget program under en selvvalgt licens, ligesom det er tilfældet med en permissiv licens. I hovedtræk gælder det kun så længe, ens program ikke er et afledt produkt af den software, som benytter GNU LGPL licensen.

2.2 Community

Et OSS community er et minisamfund, hvor omdrejningspunktet er et OSS projekt. Projektet kan inddeles i en række grupperinger. Til defineringen af disse grupperinger benyttes definitionen af (Andreev, Feller, Finnegan, & Moretz, 2010).

Passive brugere er de brugere, som har den mindste kontakt til projektet. De involverer sig som regel ikke i projektet, med mindre det omhandler support.

Aktive brugere er de brugere, som involverer sig i projektet uden at bidrage til selve udviklingen af softwaren. De kan bidrage fx i forbindelse med support eller fejlrapportering.

Perifere udviklere er de brugere, som bidrager med at skrive kode til projektet. De har typisk ikke commit⁴ adgang til projektets VCS⁵. I forbindelse med beslutninger er de som regel kun med til at give input. En perifer udvikler vil desuden blive omtalt som author, når der tales om et specifikt commit. En author er en person, som har skrevet et stykke kode, men er ikke nødvendigvis den samme person, som har udført commit.

¹ For mere information vedrørende FreeBSD licensen se <http://www.freebsd.org/copyright/freebsd-license.html>

² For mere information vedrørende GNU GPL licensen se <http://www.gnu.org/copyleft/gpl.html>

³ For mere information vedrørende GNU LGPL licensen se <http://www.gnu.org/licenses/lgpl.html>

⁴ Et commit er den proces, hvor man tilføjer kode til et VCS.

⁵ VCS (Version Control System) er det revisionssystem, der registrerer ændringer i koden, så man fx kan vende tilbage til en tidligere revision af softwaren.

Kerneudviklere er de brugere, som driver projektet. De har adgang til at committe kode til projektets VCS, og de spiller en central rolle i forbindelse med koordinering af projekt samt i beslutningsprocessen. Denne gruppering kaldes også kernegruppen.

Ledere er dem, som sætter de overordnede rammer og har det primære ansvar for projektet. Ledere kan også være en del af kernegruppen.

I denne afhandling vil aktive brugere og perifere udviklere blive anset som samme gruppering grundet måden, hvorpå et communitys størrelse vil blive opgjort. Dvs., når der refereres til det aktive community, vil det indebære begge grupperinger.

3 Problemformulering

OSS er ikke noget nyt fænomen. Der findes fx mere end 324.000⁶ OSS projekter på Sourceforge⁷. Der vil typiske blive tilknyttet flere personer til et projekt i takt med, at projektet får mere og mere succes. Dette værende en mindre gruppe kerneudviklere, som står for den primære udvikling af softwaren og typisk en langt større gruppe brugere, som i større eller mindre grad vælger at bidrage i form af patches⁸, defects⁹ eller idéer.

I forbindelse med et OSS projekt vil der skulle tages en række kritiske valg fx i form af licensvalg eller retning på projektet. Den beslutning, der bliver taget, vil få konsekvenser for projektets fremtid. Der vil derfor blive undersøgt omstændigheder omkring disse kritiske valg, samt hvilken påvirkning det vil have for et projekt. For at kunne definere et kritisk valg vil der blive brugt motivation, organisation og teknologi som afgørende faktorer. Disse faktorer vil blive benyttet i CBPP¹⁰ processen.

Til denne afhandling har det været interessant at finde en række projekter, som har befundet sig på forskellige stadier, eller er kommet til samme punkt på forskellige måder. Som udgangspunkt er der fundet tre projekter, som er inddelt i tre kategorier:

- Projektet blev startet som en enkelt persons ide og har fået en stor udviklertilslutning (Exherbo).
- Projektet blev startet af flere personer som internt projekt i en virksomhed og har opnået en vis succes men oplevede stor passivitet¹¹ fra kerneudviklergruppen (Bromine).

⁶ Se Bilag 1

⁷ For mere information henvises til www.sf.net

⁸ Patch er en fejlrettelse af kildekoden.

⁹ Defect er en identificeret fejl i programmet.

¹⁰ Common-based peer-production. Se teoriafsnit for mere information.

¹¹ Medlemmerne valgte kort forinden påbegyndelsen af denne afhandling at stoppe projektet.

- Projektet blev startet i kommercielt henseende¹² og har opnået stor succes med kommerciel support virksomhed (Varnish Cache).

For hvert af disse projekter vil det være interessant at undersøge følgende spørgsmål (både retrospektivt og nutidigt):

- Hvad motiverede opstarten af projektet?
 - Hvilke motivationsfaktorer er repræsenterede i projektet?
 - Hvilken betydning har de identificerede motivationsfaktorer haft for projektets organisation?
- Hvordan er projektet organiseret?
 - Hvordan er organisationsstrukturen?
 - Hvordan er organisationskulturen?
 - Hvordan ledes projektet?
- Hvordan påvirker den teknologiske udvikling af softwaren projektet?
 - Hvordan afstemmes de eksterne forventninger med de interne forventninger?
 - Hvordan påvirker vækst et OSS projekt?
- Hvordan påvirker følgende kritiske valg projektet?
 - Licens
 - Softwarens arkitektur
 - Distributioner
 - Ledelse
 - Kultur
 - Omverdenen
 - Når projektet skal stoppes

Når ovenstående spørgsmål er blevet besvaret for hvert enkelt projekt, vil det være interessant at undersøge følgende spørgsmål:

- Hvilke typiske kritiske valg konfigurationer findes der i OSS projekter?
- Kan OSS projekternes kritiske valg og deres løsninger anvendes til vejledning af iværksættere (generaliseres til iværksætter teori)?

¹² Det var fra starten af projektet besluttet, at softwaren skulle udgives som OSS.

4 Emneafgrænsning

I open source er der en lang række elementer, som man typisk vil knytte til ordene open source. I dette afsnit redegøres for, hvorfor en række elementer ikke er inddraget.

4.1 Aktive community

Det aktive community spiller en væsentlig rolle i et OSS projekt, men selvom denne gruppering spiller en central rolle i OSS projektets succes, er det valgt ikke at inkludere dem i denne afhandling. Denne beslutning er bygget på, at et OSS projekt drives af kernegruppen. Det er denne gruppering, som har mandat til at træffe en beslutning, når et kritisk valg opstår. Typisk er det også denne gruppering, som besidder magten over, hvordan projektet ledes, og derfor kan træffe beslutning uden at inddrage resten af communityet. Dermed ikke sagt, at det aktive community ikke har indflydelse. Det vil det ofte have, men i sidste instans vil det være kernegruppens beslutning, som vil bane vejen for projektets fremtid.

I forbindelse med udmålingen af et communityets størrelse vil den aktive del af community blive inddraget ved at undersøge projektets VCS samt fejlrapporteringssystem. Andre steder, hvor det er relevant, er communityet inddraget set ud fra kernegruppens perspektiv.

4.2 Komplementære teknologier

Det er valgt ikke at tage komplementære teknologier som revisionssystemer, hjemmesider, fora og lignende med i denne afhandling, da disse teknologier er let tilgængelige og ikke kan ses som en konkurrencemæssig faktor. Tværtimod kan det anses som en ulempe, hvis man ikke udnytter denne teknologi, og det vil derfor være attraktivt at anvende. Ydermere er disse komplimenterende teknologier frit tilgængelige. Der findes flere portaler¹³, som giver adgang til alle disse teknologier uden omkostning for OSS projekter. Der, hvor det vurderes nødvendigt, vil der være referencer til disse systemer uden at gå i dybden med dem.

4.3 Internet

Internettet er en central del i distributionen af selve softwaren og kommunikationen imellem udviklere og brugere. Målet med afhandlingen er ikke at undersøge internettets betydning for OSS eller kommunikationsformen, men at undersøge de underliggende emner som motivation, organisation og teknologi i forbindelse med kritiske valg. Det er antaget, at de fleste af de personer, som er involveret i open source, om det er kerneudviklere eller brugere, er teknologiske kompetente, når det kommer til brug af internettet. Derfor vil internettet i denne afhandling blive taget for givet.

¹³ Fx www.sf.net og www.assembla.com

4.4 Internationalisering (i18n)

I takt med at projekter vokser, vil sammensætningen af udviklere og brugere blive mangfoldig. Det vil sige, at der vil være personer fra mange forskellige lande, som kunne være interesseret i en lokal version af softwaren. Dette emne er blevet fravalgt, da det er en underlæggende del i selve arkitekturen af softwaren. Arkitekturs valg er anset som et kritisk valg, da arkitekturen vil være med til at sætte sit præg på, hvordan softwaren vil udvikle sig. I18n er derfor ét i blandt mange afledte produkter af arkitekturen, men målet med afhandlingen er ikke at identificere mulige produkter eller produktets påvirkning af projektet. Derimod er det interessant at se på, hvordan arkitekturen får en afgørende betydning på motivationen og organisationen, da en arkitektur kan fostre mange anonyme brugere som kontribuerer kode, eller omvendt kræver en mere dedikeret kerneudviklergruppe.

4.5 Open source historiske bølge

Open source har udviklet sig meget gennem tiden. I begyndelsen af softwarens udvikling blev software delt på tværs af organisationer og universiteter, hvor softwaren blev anvendt frit uden licensbegrænsninger. Gennem 1980'erne kom bruddet med software uden licensbegrænsninger, og derved opstod proprietær software. I dag findes der utallige open source projekter og open source licenser.

Teknologisk set er der også sket meget, men det har ikke været hensigten hverken at prøve at forudse, hvor open source er på vej hen, eller hvad den næste teknologiske bølge bliver inden for open source. Det er derfor valgt at have det teknologiske perspektiv med, men det skal ses i forbindelse med selve softwarens udvikling over tid. Denne udvikling kan have implikationer for, hvorledes grænsefladerne imellem kerneudvikler og bruger kan få indflydelse på projektet og de kritiske valg, der kan opstå inden for open source.

4.6 Omgivelser

Der er ikke set på omgivelserne omkring et open source community, hverken konkurrencemæssig eller hvorvidt man kan retfærdiggøre et projekt set i forhold til den store mængde software, som allerede er tilgængeligt som open source. Det er antaget, at et projekt, der er etableret og har fået opbygget et community, har haft et behov for netop den funktionalitet, som projektet har implementeret. Det kunne være en mindre ændring i forhold til allerede eksisterende projekter eller et nyt grafisk interface, men det har ikke været relevant i denne sammenhæng.

4.7 Kritiske valg

I problemformuleringen er der udpeget syv generiske kritiske valg. Dette skal ikke forstås som en afgrænsning fra andre kritiske valg. Disse valg har været resultatet af tre elementer; forfatterens involvering i Bro-mine, teori omhandlende OSS projekter, og at valgene kunne konkluderes som værende kritiske i de under-

søgte projekter. Oprindeligt blev der oplistet en række kritiske valg, som kun reflekterede forfatterens erfaring samt teori. Disse var:

- Hvis projektet skal forkes
- Hvordan rekrutteres der nye kerneudviklere?
- Licensvalg
- Rettighedsstruktur fx hvem har rettigheder til at skrive til projektets VCS.
- Udviklingspolitikker, herunder retningen på projektet
- Arkitektur, tekniske
- Distributioner
- Forretningsmodel, herunder også finansiering.

Datagrundlaget har ikke betegnet alle ovenstående valg som værende kritiske, hvorved det har været nødvendigt at generere en række generiske kritisk valg som styringsredskab for denne afhandling.

5 Videnskabeligmetode

I dette afsnit er der redegjort for, hvilket paradigme afhandlingen vil tage udgangspunkt i, samt den videnskabelige metodes opbygning.

5.1 Paradigme

Der er taget afsæt i Roy Bhaskars kritiske realisme. Bhaskar har siden begyndelsen af kritisk realisme¹⁴ videreudviklet konceptet kritisk realisme. Der er taget afsæt i første bølge inden for kritisk realisme, som er fra før 1993. I 1993 udgav Bhaskar bogen "*Dialectic: The pulse of freedom*", som var begyndelsen på en ny bølge inden for kritisk realisme. Denne nye form for kritisk realisme fik navnet dialektisk kritisk realisme. I denne bølge var det filosofien i bred forstand, som var i fokus¹⁵. Som nævnt er der valgt at tage udgangspunkt i den første bølge, hvor det primære formål med kritisk realisme var at understøtte og omskabe videnskabelig praksis¹⁶.

For at forklare, hvad kritisk realisme er, er der taget udgangspunkt disse tre spørgsmål "1: *Ontological: What is the nature of the "knowable"? Or, what is the nature of "reality"?* 2: *Epistemological: What is the nature of the relationship between the knower (the inquirer) and the known (or knowable)?* 3: *Methodological: How should the inquirer go about finding out knowledge?*" Citat (Guba, 1990, s. 18)

¹⁴ I 1975 udgav Bhaskar "Transcendental realisme", som var startskuddet til kritisk realisme (Buch-Hansen & Nielsen, 2005) s. 7

¹⁵ (Buch-Hansen & Nielsen, 2005, s. 92)

¹⁶ (Buch-Hansen & Nielsen, 2005, s. 92)

Udover at tage afsæt i ovenstående spørgsmål er det vigtigt at forstå, at inden for kritisk realisme, er verden niveaudelt. Bhaskar inddeler verden i tre domæner¹⁷:

- Det empiriske domæne
- Det faktiske domæne
- Det virkelige domæne

Disse domæner kan groft oversættes til henholdsvis ontologi, epistemologi og metodologi. Denne sammenhæng er redegjort for under hvert element, som er tilknyttet Gubas tre spørgsmål.

5.1.1 Epistemologi

Indenfor kritisk realisme betegnes epistemologi, viden om verden, som den transitive dimension. I kritisk realisme er epistemologien subjektiv¹⁸. Epistemologien bliver subjektiv, idet vi ikke kan se verden, som den virkelig er eller fungerer uden at tillægge den et værdisæt¹⁹. Ved at der tillægges et værdisæt, anerkendes det, at al viden er produceret i en social kontekst. Det er denne sociale kontekst, som ligger til grund for den enkeltes værdisæt være det teori, modeller, analysemetoder osv. Det, at vores viden, de transitive objekter, er produceret i en social kontekst, gør den konstrueret og fejlbarlig.

Vores viden om verden er derfor alt det, som vi erfarer og observerer. Dette er, hvad Bhaskar betegner for det empiriske domæne.

5.1.2 Ontologi

Indenfor kritisk realisme betegnes ontologien, viden om noget, som den intransitive dimension. Ontologien bygger på kritisk realisme. De intransitive objekter, som ligger til grund for videnskabelig undersøgelse i den intransitive dimension, eksisterer uafhængig af vores viden om dem, hvilket underbygger realismen i kritisk realisme. Den kritiske del består i, at undersøgeren skal være kritisk overfor den undersøgelse, der bliver udført, fordi det er umuligt for mennesket at opnå fuldstændig viden om de intransitive objekter, da de menneskelige sensoriske og intellektuelle mekanismer ikke er perfekte²⁰.

Vores viden om noget kommer derfor til udtryk gennem begivenheder og fænomener, som Bhaskar betegner det faktiske domæne.

¹⁷ (Buch-Hansen & Nielsen, 2005) s. 24

¹⁸ (Nygaard, 2005) s. 25

¹⁹ (Guba, 1990) s. 24.

²⁰ (Guba, 1990) s. 20

5.1.3 Metodologi

Metodologien indenfor kritisk realisme bygger på transformation og dialog. Transformation i kritisk realisme skal forstås i samspillet mellem undersøger og det undersøgte. I dette samspil er det hensigten at transformere den virkelige verden, så vi højner vores bevidsthed og derved eliminerer den falske bevidsthed, som udspringer fra en misforståelse af virkeligheden. Gennem dialog er det den kritiske realists mål at eliminere den falske bevidsthed ved at samle folk omkring en måske sand og fælles bevidsthed²¹.

For at finde den sande bevidsthed skal undersøgeren undersøge, hvilke strukturer, kræfter og mekanismer, der konstituerer de intransitive objekter. Det vil sige, at fokus flyttes fra de observerbare fænomener og begivenheder til de underliggende strukturer, kræfter og mekanismer, som understøtter eller forårsager disse fænomener og begivenheder²². Dette niveau betegner Bhaskar som det virkelige domæne.

5.2 Forskningsprocessen

Til denne forskningsproces er der taget udgangspunkt i to forskellige forskningsprocesser af henholdsvis Claus Nygaard²³ og John Mingers²⁴. Nygaards femskridtsforskningsproces skal bruges som hovedelementer i denne forskningsproces. De fem skridt er "*1: Klarlægge formålet med forskningen. 2: Lokalisering og intuitiv karakteristik af fænomenet. 3: Empiriske erfaringer af fænomenet. 4: Abduktion af de konstituerende mekanismer. 5: Præsentation af den hypotetiske konklusion vedrørende konstituering af fænomenet*" Citat (Nygaard, 2005, s. 54)

Denne forskningsproces kan henledes til vandfaldsmodellen, som står delvist i kontrast til Mingers fire faser, som ikke kan ses som deciderede skridt men mere som en iterativ model, hvor man i større eller mindre grad benytter sig af de forskellige faser gennem hele forskningsprocessen. De fire faser er defineret som "*Appreciation (What is happening?). Analysis (Why is it happening?). Assessment (How could the situation or explanation be different?). Action (And so what?)*" citat (Fleetwood & Ackroyd, 2004, s. 154-185).

Ved at kombinere disse to forskningsprocesser er der opnået en opbygning, som er klarlagt ved, at man skal forholde sig til de fem identificerede skridt af Nygaard. Samtidig skal man forholde sig kritisk til, hvad der foretages i denne afhandling og løbende evaluere ud fra de fire identificerede faser af Mingers.

Da forfatteren har været involveret i et OSS projekt²⁵, er denne afhandling baseret på evidensforskning. Dette skyldes, at forfatteren via sit virke i projektet har erfaret, at et OSS projekt vil stå overfor en række

²¹ (Guba, 1990) s. 24

²² (Buch-Hansen & Nielsen, 2005) s. 24-25

²³ (Nygaard, 2005) s. 54-56

²⁴ (Fleetwood & Ackroyd, 2004) s. 184-187

²⁵ Forfatteren har i perioden september 2007 til januar 2012 været involveret i Bromine.

kritiske valg. Derfor er det valgt at afprøve en tese om, at OSS projekter står overfor en række kritiske valg, som kan generaliseres ved at kombinere forfatterens egne erfaringer med teori udarbejdet af andre forskere indenfor OSS.

6 Den videnskabeligmetodes betydning for undersøgelsen

I dette afsnit er det konkretiseret, hvilken betydning det videnskabelige paradigme har fået for denne afhandling. Hovedopbygningen har fulgt Nygaards fem skridt. Der er i det følgende redegjort for hvert skridt i processen. Der er implicit blevet gjort brug af Mingers fire faser for at holde et kritisk perspektiv til afhandlingen under hele forløbet.

6.1 Formålet med forskningen

Formålet med forskningen har været at skabe ny viden omkring de valg, som et OSS projekt kan komme til at stå overfor, og som vil have en væsentlig påvirkning på projektet. Udgangspunktet har været den sociale kontekst, hvorpå et OSS projekt bliver sammensat, og hvad der sker, når et projekt bliver påvirket i væsentlig grad. Ved identificering af disse valg, samt hvorledes de kan påvirke et projekt, har det været hensigten at skabe opmærksomhed omkring de underliggende mekanismer, som har konstitueret et givent fænomen, når et projekt har stået over for et kritisk valg. Ved at skabe opmærksomhed omkring disse underliggende mekanismer har det givet de implicerede personer i et givent projekt et bedre beslutningsgrundlag. Dette beslutningsgrundlag har kunnet tage højde for valgets betydning for både projektet og dets deltagere.

6.2 Lokalisering og intuitiv karakteristik af fænomenet

I dette skridt er de kritiske valg samt den sociale kontekst, hvorpå at OSS projekter bliver konstitueret, blevet lokaliseret og karakteriseret. Som kritisk realist eksisterer disse konstellationer af valg og sociale relationer uafhængig af, om de erkendes eller ej, som intransitive objekter. Der er ikke tvivl om, at de eksisterer, da man empirisk kan erfare, at de findes ved at gå ind på fx www.sf.net og opnå adgang til en stor mængde OSS projekter. Udover forfatterens egne tidligere empiriske erfaringer inden for OSS, er der taget udgangspunkt i tidligere undersøgelser, som har haft til hensigt at udforske, hvilke udfordringer og muligheder der kan opstå i OSS projekter. Dette har gjort det muligt at udarbejde en intuitiv karakteristik af fænomenet, som er brugt til at tage afsæt i, når der til denne afhandling har skullet indsamles empiriske erfaringer af fænomenet. Endvidere har det givet en mulighed for at give et bud på, hvor det kunne være interessant at foretage de empiriske erfaringer.

6.3 Empiriske erfaringer af fænomenet

I denne del af forskningsprocessen er der gjort empiriske erfaringer af fænomenet. Typisk er der brug for kvalitative data for at få viden omkring sammenhængen mellem fænomener og de underliggende mekanismer²⁶. Som kritisk realist er man ikke tvunget til at arbejde med én bestemt dataindsamlingsteknik. Man har også muligheden for at bruge kvantitative data. Kvalitative data har i denne sammenhæng, hvor der undersøges sociale relationer og påvirkningen af bestemte valg, været at foretrække frem for kvantitative data. Dette er grundet, at gennem dialog med respondenter har det været muligt at finde frem til de underliggende mekanismer, som har udgjort den sociale kontekst, de har befundet sig i. Samt hvordan de har forholdt sig til de kritiske valg, også selvom de umiddelbart ikke selv har været klar over, hvorfor et givent fænomen har været konstitueret, som det var.

For at filtrere subjektiviteten af de empiriske erfaringer er der blevet benyttet metodetriangulering (Andersen, 2005, s. 164). Det har givet en bedre forståelse af de intransitive objekter, hvorved den subjektive indflydelse er blevet reduceret, når man har gjort sig sine empiriske erfaringer.

6.4 Abduktion af de konstituerende mekanismer

Efter at empiriske erfaringer af fænomenet er blevet indsamlet, skal de underliggende strukturer, kræfter og mekanismer, der har konstitueret de intransitive objekter, abduces. Abduktionen har gjort det muligt at generere ny viden om det virkelige domæne enten via en logisk eller en intuitiv hypotetisk konklusion.

I denne afhandling er det forsøgt, at på- eller afvise tesen om, at man har kunnet generalisere kritiske valg inden for OSS. Grundet afhandlingens afgrænsede datagrundlag på tre projekter vil konklusionen kun kunne bruges som en indikation på, om der findes generelle kritiske valg indenfor OSS. Derved er grundlaget opbygget for et større studium, hvor et større datagrundlag vil kunne give et bedre billede af, om det er muligt at generalisere disse kritiske valg.

6.5 Præsentation af den hypotetiske konklusion vedrørende konstituering af fænomenet

Det sidste skridt i denne forskningsproces har været at præsentere den hypotetiske konklusion for personer, som er involveret i OSS og andre forskere. Det er valgt, at denne afhandling skal danne grundlaget for dialogen imellem forskere og medlemmerne af OSS samfundet. Dette har betydet, at der ikke kan redegøres for, hvorvidt det har været muligt at skabe transitive objekter i forbindelse med udarbejdelsen. Ved at gøre denne afhandling genstand for dialog har det være muligt, i det omfang at datagrundlaget tillader det,

²⁶ (Nygaard, 2005) s. 55

at være med til at be- eller afkræfte hypotesen. Hvis den hypotetiske konklusion kan vinde accept og anvendelighed hos de involveret personer i OSS samt forskermiljøet, har det skabt transitive objekter²⁷.

7 Teori

For at belyse OSS projekters kritiske valg, er der taget udgangspunkt i en række teorier, som hver især omhandler de områder, der har været ønsket at undersøge. De artikler, som har behandlet mere end ét område, er blevet delt op i forskellige afsnit, fx er der brugt en artikel, der har behandlet både motivation og organisation. Artiklens indhold er derfor blevet delt op i to afsnit, som hver har beskrevet deres respektive teoriområde. Ved denne opdeling har det anskueliggjort begrebsapparatet ved at konsolidere forskellige artikler, som er blevet benyttet på empirien.

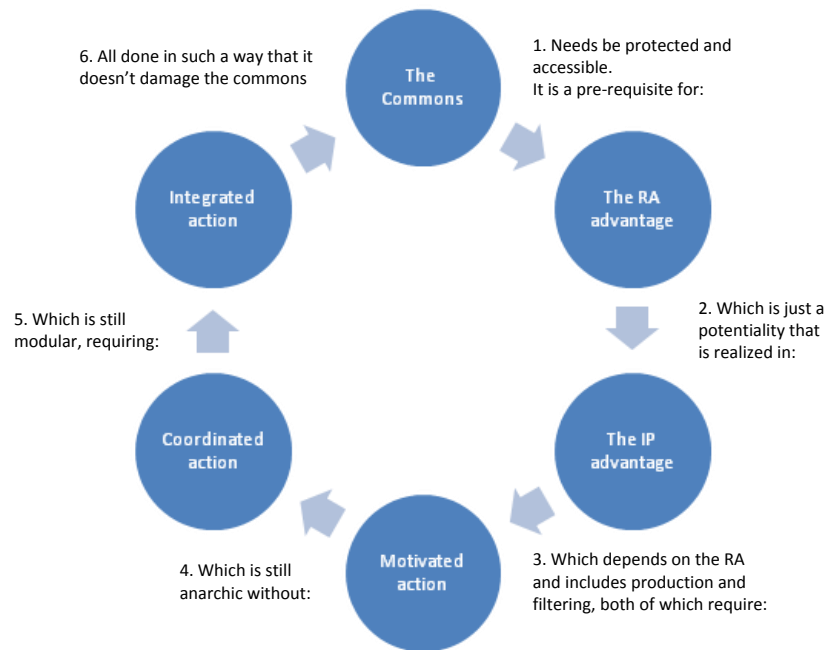
7.1 CBPP processen

Commonsbased peerproduction (CBPP) er en teori, som tager udgangspunkt i, at der er nogle muligheder og udfordringer i et OSS projekt. Artiklen, der tagers afsæt i, er *"Conceptualizing The Commonsbased Peerproduction Of Software: An Activity Theoretic Analysis"* (Andreev, Feller, Finnegan, & Moretz, 2010).

Som udgangspunkt til, hvilke udfordringer og fordele der ligger i at arbejde med OSS, er der taget udgangspunkt i *"Conceptualizing The Commonsbased Peerproduction Of Software: An Activity Theoretic Analysis"*. Denne artikel omhandler, hvordan man kan overkomme de primære udfordringer (motivation og organisation), samt hvordan OSS projekter kan drage nytte af oplagte muligheder. Disse muligheder udspringer fra den måde, et OSS projekt kan sammenlignes med en basar lignende struktur²⁸. Artiklen har konceptualiseret et afhængighedsdiagram, som sætter fordele i relation til en række udfordringer. Det skitseres i Figur 2.

²⁷ (Nygaard, 2005) s. 57

²⁸ (Andreev, Feller, Finnegan, & Moretz, 2010) s. 3



Figur 2 Proces diagram "flow of peer production process"²⁹

Figuren kan kort beskrives som³⁰.

1. Projektet skal være beskyttet og tilgængelig, fx ved valg af en GPL licens og have et offentlig VCS, så andre kan få adgang til projektet.
2. Ressourceallokeringsfordelen er ikke givet, men en mulighed som er afhængig af, hvor godt projektet trives.
3. Ovenstående mulighed er realiseret gennem informationsbehandlingsfordelen, som er afhængig af en bred deltagelse og en effektiv filtrering af kontributioner.
 - a. For at få en bred deltagelse kræver det, at man kan overkomme motivationsudfordringen ved at udnytte de forskellige, skalerbare og kontekst bestemte motivationsfaktorer.
4. Frem til nu vil processen være præget af anarki.
 - a. Dette anarki kan kun overkommes ved at tage fat på de organisatoriske udfordringer ved hjælp af koordinering.
5. Indtil nu vil processen stadig være modulær.
 - a. Før man kan få fuldt udbytte af informationsbehandlingsfordelen kræves det, at man kan vælge, hvilke løsninger der skal bruges, og hvordan de skal integreres i projektet.
6. Ovenstående skal udføres sådan, at det ikke skader projektet.

²⁹ Kilde: Billede og tekst (Andreev, Feller, Finnegan, & Moretz, 2010, s. 5)

³⁰ (Andreev, Feller, Finnegan, & Moretz, 2010, s. 5)

Denne artikel bearbejder kun overfladisk med emnerne motivation, kultur og organisation. Derfor er der taget udgangspunkt i de motivationsfaktorer, der er blevet henvist til, og fundet relevant teori, som har kunnet anvendes i forbindelse med indsamling og analyse af empirien. I forbindelse med kultur er der ikke henvist specifikt til nogen teori. Derfor er det valgt at benytte en funktionalistisk tilgang til kulturanalyse. Vedrørende organisation henvises til basarorganisationen, hvorved der er suppleret med teori, der specifikt omhandler arbejdsdeling i forhold til basar- og katedralorganisationer.

7.2 Motivation

Inden for OSS er motivation nævnt som en af de primære udfordringer, et projekt kommer til at stå overfor³¹. For at undersøge, hvad der motiverer folkene bag OSS, er der taget udgangspunkt i tidligere undersøgelser, som har haft til hensigt at undersøge motivationen bag samme gruppering.

Der er udvalgt en artikelsamling, *"Perspectives on Free and Open Source Software"* (Feller, Fitzgerald, Hissam, & Lakhani, 2005). Denne artikelsamling er redigeret men ikke nødvendigvis skrevet af nævnte forfattere. Artikelsamlingen er anvendt som det teoretiske grundlag for analyse af motivationsfaktorer.

7.2.1 Motivationsudfordringen

I (Andreev, Feller, Finnegan, & Moretz, 2010) gives tre måder, hvorpå et OSS projekt kan overkomme de udfordringer, der kan opstå i forbindelse med motivation.

7.2.1.1 Tage hensyn til forskellige motivationsfaktorer

En af de største fordele ved OSS er, at man kan drage nytte af flere forskellige motivationsfaktorer. Disse motivationsfaktorer defineres som indre og ydre motivation herunder monetær godtgørelse. Ved at kunne drage nytte af flere typer motivationsfaktorer i større arbejdsammenhæng, kan man reducere motivationsproblemet til et samarbejdsproblem³².

Der er redegjort for definitionen af disse motivationsfaktorer i henholdsvis afsnit 7.2.2 og 7.2.3, som specifikt omhandler indre og ydre motivation herunder et økonomisk perspektiv på motivation i OSS projekter.

7.2.1.2 Skalerbar motivation

Skalerbar motivation hentyder til softwarens arkitektur. Hvis arkitekturen bygger på små moduler, vil det være muligt for deltagerne i projektet at arbejde med en lille del af systemet isoleret fra resten af systemet, og derved bidrage til projektet med en relativt lille indsats. Hvis man alternativt har et system, som ikke er modulopbygget, kan der være en høj indlæringskurve, da systemet kan være stort og komplekst. Artiklen

³¹ (Andreev, Feller, Finnegan, & Moretz, 2010) s. 1

³² (Andreev, Feller, Finnegan, & Moretz, 2010) s. 9

nævner dette som en betydelig fordel, at man kan drage nytte af personer, som måske ikke føler sig særligt motiverede til at bidrage til projektet³³. På grund af at der kræves en relativt lille indsats for at bidrage, vil man føle, at den motivation de har om end lille, vil være grundlag nok for deres bidrag til projektet.

Andre opgaver kræver dog en væsentlig større indsats. Her nævner artiklen opgaver, der måske ikke vil være særligt incitament til at udføre, men måske vil blive udført af personer, som får en monetær godtgørelse for deres indsats. Artiklen indikerer dog, at der kan opstå et skæl imellem de deltagere, der får betaling, og de, der ikke gør.

Man kan udover de to ovennævnte muligheder, modulbaseret og monetær godtgørelse, også drage nytte af roller, som skal være med til at motivere deltagerne. I nogle OSS projekter hersker et hierarki, som bygger på meriter. Fx bygger Apache Software Foundation³⁴ (ASF) på denne type merit, som betegnes meritocracy³⁵. For at avancere i dette hierarki kræver det, at man deltager aktivt og opnår merit hos de andre deltagere. Denne anerkendelse kan virke som en motivationsfaktor.

7.2.1.3 Kontekstbestemt motivation

Kontekstbestemt motivation referer til den type motivation, som er baseret på sociale relationer, hvor personerne føler, at de identificerer sig med en bestemt gruppe eller fællesskab. Denne type motivation relaterer sig til forpligtigelses og fællesskabsbaseret indre motivation, som behandles i afsnit 7.2.2.

7.2.2 Indre motivation

Til at definere indre motivation tages udgangspunkt i "*Why Hackers do what they do: understanding motivation and effort in free/open source software projects*"³⁶. I denne artikel skelnes der mellem to typer motivation, henholdsvis indre og ydre motivationsfaktorer. De indre motivationsfaktorer defineres, når man udfører en aktivitet for den tilfredsstillelse, man opnår ved udførelsen set i kontrast til, hvis man fx fik en modydelse for at udføre en given ydelse³⁷. De indre motivationsfaktorer kan endvidere deles op i henholdsvis glædesbaseret og forpligtigelses- og fællesskabsbaserede motivationsfaktorer.

Den glædesbaserede indre motivation karakteriseres som værende den glæde eller tilfredsstillelse, som personen opnår ved at udføre en given aktivitet. I artiklen postuleres det, at der findes et ekvilibrium, hvorved personen opnår maksimal glæde. Dette opnås ved, at aktiviteten passer til personens færdighedsniveau. Hvis fx aktivitetens færdighedsniveau overstiger personens reelle evner, vil det fremprovokere angst, og

³³ (Andreev, Feller, Finnegan, & Moretz, 2010) s. 9

³⁴ For mere information omkring Apache projektet se <http://www.apache.org>

³⁵ (Andreev, Feller, Finnegan, & Moretz, 2010, s. 10)

³⁶ Artiklen indgår i (Feller, Fitzgerald, Hissam, & Lakhani, 2005)

³⁷ (Feller, Fitzgerald, Hissam, & Lakhani, 2005) s. 4

hvis personens evner overstiger aktivitetens færdighedsniveau, vil personen opleve aktiviteten som ked-sommelig. I forbindelse med OSS kan dette sidestilles med, hvilke opgaver en given udvikler ønsker at påtage sig. Herved prøver personen at vælge opgaver, der matcher personens færdigheder³⁸.

Den forpligtigelse og fællesskabsbaserede indre motivation karakteriseres som værende de faktorer, som gør, at en person vil føle sig forpligtiget eller handle ud fra fællesskabets normer til at udfører en given aktivitet. Set i relation til OSS vil mange personer, som vælger at deltage i et OSS projekt, have en stærk følelse af at tilhøre en gruppering. Denne gruppering kan være præget af specielle handlingsmønstre, som kan findes i kulturens normer og værdier³⁹, hvilket vil få den pågældende person til at føle en stærkere træng til at påtage sig en given aktivitet.

7.2.3 Ydre motivation

Til definitionen af ydre motivation tages der udgangspunkt i "*Economic perspectives on open source*"⁴⁰. Denne artikel udsprang i et økonomisk perspektiv. Artiklen fremlægger en udviklers deltagelse i et projekt, både kommercielt og open source med begrundelsen, at personen skal ende ud med en nettofortjeneste. Denne nettofortjeneste skal beregnes som⁴¹:

$$\begin{aligned} \text{Nettofortjeneste} \\ &= (\text{direkte fortjeneste} - \text{direkte omkostning}) \\ &+ (\text{forsinket fortjeneste} - \text{forsinket omkostning}) \end{aligned}$$

Artiklen beskriver en række forskellige fortjenester og omkostninger. Når man vælger at deltage i et OSS projekt, fravælger man direkte eller indirekte andre muligheder, som man ellers kunne have brugt sin tid på. Dette betegnes for en nuværende omkostning, da man ved at arbejde på OSS projektet allokerer en mængde af ens begrænsede tid, som ellers kunne være brugt på noget andet. En forsinket omkostning kunne være, når man bruger tid på et OSS projekt i sin arbejdstid, så bliver ens andre arbejdsopgaver forsinkede. En direkte fortjeneste, som kunne modvirke disse omkostninger, kunne være ved at øge ens produktivitet ved, at man rettede nogle fejl eller lavede ændringer i et OSS projekt, som man kunne inddrage i sit arbejde. Forsinket fortjeneste beskrives som incitamenter vedrørende karriere eller ego tilfredsstillelse. Disse to typer incitamenter betegnes som signaleringsincitamenter. Karriereincitamenter vedrører fremtidige jobmuligheder eller fremtidig adgang til venture kapitalmarkedet⁴². Egotilfredsstillelsesincitamenter vedrører anerkendelse af ligemænd. Disse signaleringsincitamenter er stærkere, når "*The more visible the*

³⁸ (Feller, Fitzgerald, Hissam, & Lakhani, 2005) s. 5

³⁹ (Feller, Fitzgerald, Hissam, & Lakhani, 2005) s. 5-6

⁴⁰ Artiklen indgår i (Feller, Fitzgerald, Hissam, & Lakhani, 2005)

⁴¹ (Feller, Fitzgerald, Hissam, & Lakhani, 2005) s. 57

⁴² (Feller, Fitzgerald, Hissam, & Lakhani, 2005) s. 58

performance to the relevant audience (peers, labor market, venture capital community). The higher the impact of effort on performance. The more informative the performance about talent" citat fra (Feller, Fitzgerald, Hissam, & Lakhani, 2005, s. 58)

I artiklen sammenlignes OSS og kommercielle projekters incitament. Der tages afsæt i delen om open source. Forfatteren identificer to muligheder til, hvordan et open source projekt kan reducere omkostninger for udvikleren henholdsvis:

- Alumni effekten, fx når der undervises i programmet på undervisningsinstitutioner.
- Den værdi, der skabes ved brugertilpasning og fejlrettelser.

Når det kommer til signaleringsincitament har (Feller, Fitzgerald, Hissam, & Lakhani, 2005) identificeret tre grunde til, hvorfor open source projekter kan stå stærkere end kommercielle projekter.

- Bedre performancemåling set i forhold til gennemslagskraften i OSS.
- Fuld initiativ, dvs., at det er op til den enkelte udvikler, hvad han vil beskæftige sig med.
- Større flygtighed på open source markedet end det traditionelle arbejdsmarked.

7.3 Organisation

For at belyse et projekts organisation er der taget udgangspunkt i organisationskultur og -struktur. En kulturanalyse er vigtig, idet den forsøger at finde frem til, hvad der driver organisationen i forhold til dens eksterne tilpasning og interne integration. Dette bliver manifesteret i form af artefakter, værdier og grundlæggende antagelser. "*Kultur giver indsigt i irrationelle og "tilsyneladende meningsløse ting i organisationer, som har indflydelse på, hvorfor mennesker tænker, føler og handler som de gør"* (Schein, 196:11)." citat fra (Schultz, 1990, s. 27).

Organisationsstrukturen kan yderligere supplere forståelsen af organisationen, da organisationsstrukturen er med til at påvirke adfærden i forhold til stabilitet, begrænsninger og koordination⁴³.

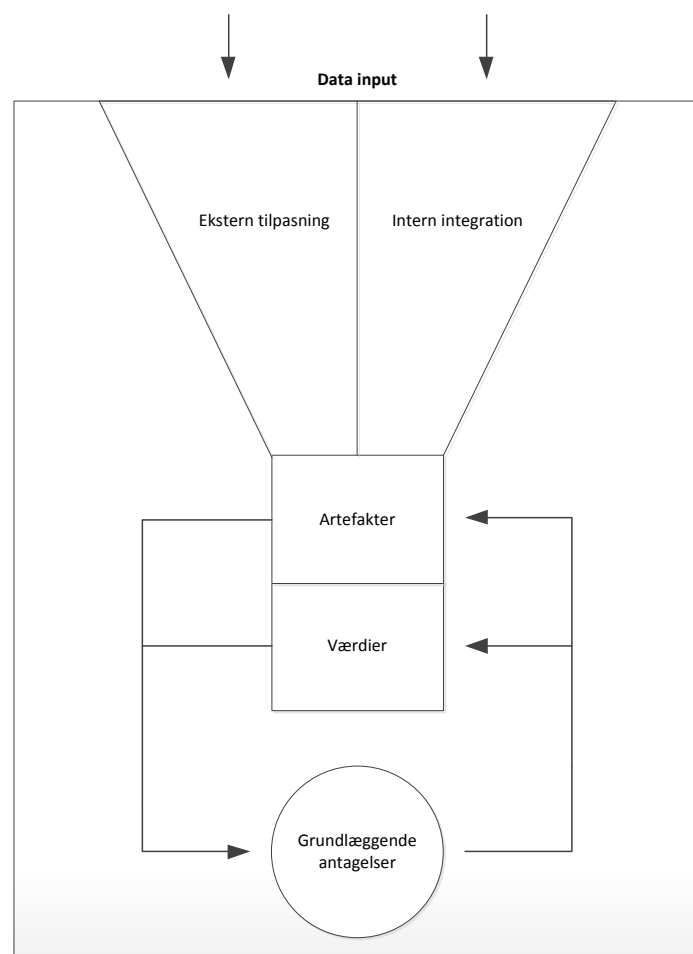
7.3.1 Organisationskultur

For at kortlægge et projekts organisationskultur er der taget udgangspunkt i Majken Schultz (Schultz, 1990). Hendes teori er bygget på Edgar Schein's funktionalistiske kulturanalyse.

Modellen, som er brugt i forbindelse med analysen, er skitseret i form af Tragten⁴⁴.

⁴³ (Jacobsen & Thorsvik, 2002) s. 70

⁴⁴ (Schultz, 1990) s.31



Figur 3 Tragten⁴⁵ gengivet

Ved Tragten skal forstås, at organisationen har to problemstillinger i forhold til dens overlevelse. Ekstern tilpasning, som omhandler, hvordan organisationen tilpasser sig sine omgivelser, og intern integration, som omhandler, hvordan organisationen integrerer sine interne processer. Disse to problemstillinger skal sættes i relation til de tre analyseniveauer; artefakter, værdier og grundlæggende antagelser.

Artefakter inddeles i fem kategorier; fysiske udtryk, sprog, historier, teknologi og synlige traditioner⁴⁶. Artefaktniveauet er et udtryk for, hvad der umiddelbart kan observeres.

Værdier defineres som værende det, medlemmerne af organisationen siger, men ikke nødvendigvis det de gør. Værdier kan derfor være modstridende ved at medlemmer handler på én måde og fortæller noget andet.

⁴⁵ (Schultz, 1990) s. 31

⁴⁶ (Schultz, 1990) s. 32

Grundlæggende antagelser kan ikke umiddelbart artikuleres uden analyse. De grundlæggende antagelser udgør det kulturelle paradigme. Ved at kortlægge mønstre af grundlæggende antagelser er det muligt at tilbageføre de grundlæggende antagelser til værdierne og artefakterne. Tilbageføring vil kunne give artefakter og værdier mening, som ellers "*kan forekomme irrationelle og meningsløse elementer i den organisatoriske forvirring*" (Schultz, 1990, s. 29).

Metoden, Tragten, består i at artefakter og værdier bliver sorteret i deres respektive funktionsområde inden for ekstern tilpasning og intern integration. Hvert funktionsområde vil blive bearbejdet isoleret fra de andre funktionsområder. De indsamlede data i form af sorterede artefakter og værdier bliver efterfølgende genstand for analyse, hvor man forsøger at finde frem til organisationens grundlæggende antagelser. De grundlæggende antagelser danner derefter fundamentet for at forstå artefakterne og værdiernes funktion i organisationen. Kulturparadigmet dannes ud fra de grundlæggende antagelser og skal efterfølgende tilbageføres til artefakterne og værdierne. Det er i denne proces, at det er muligt at forklare artefakterne og værdierne. Sidste del af analysen er fravalgt, hvor de grundlæggende antagelser tilbageføres. Dette skyldes, at målet med kulturanalysen har været at finde frem til det kulturelle paradigme, som vil være et af de elementer, som vil kunne forklare, hvorfor et givet valg kan betegnes som kritisk. Det har derfor ikke været væsentligt for denne afhandling, at samtlige artefakter og værdier bliver forklaret.

Til operationalisering af modellen er der taget udgangspunkt i Scheins fem funktionsområder vedrørende ekstern tilpasning og operationalisering af Scheins seks funktionsområder vedrørende intern integration hos (Schultz, 1990). Disse er reduceret til fem i operationalisering hos (Schultz, 1990). De ti funktionsområder er defineret som:

<i>Ekstern tilpasning</i>	<i>Intern integration</i>
<i>Mission og strategi</i>	<i>Fælles sprog og begrebskategorier</i>
<i>Mål</i>	<i>Gruppegrænser og kriterier for optagelse og udelukkelse</i>
<i>Midler</i>	<i>Magt og status</i>
<i>Måling</i>	<i>Ledelse</i>
<i>Korrektion</i>	<i>Standarder</i>

Citat fra (Schultz, 1990) s.30-40

I tråd med nedenstående citat fra (Schultz, 1990) er der fremover arbejdet med en kombination af funktionsområderne hos Scheins og (Schultz, 1990). *"I denne bog sorteres og omformuleres kulturens funktioner ud fra en generel organisationsanalyse[...]. Men den generelle opdeling i ekstern tilpasning og intern integration fastholdes. Hermed fraviges Scheins påstand om, at de definerede overlevelseshproblemer er lige relevante for ethvert socialt system."* citat fra (Schultz, 1990, s. 37).

7.3.1.1 Udfordringer

Den funktionalistiske kulturanalyse fremhæver værdier som værende af normativ karakter⁴⁷. Dette betyder, at respondenternes udtalelser om et givent emne kan relateres til Scheins definition af værdier, og det kan gøres til genstand for diskussion. I forbindelse med OSS er der ikke nødvendigvis en fælles fysisk arbejdsplads, og arbejdet kan derfor blive udført distribueret og asynkront.

Det har gjort observation af respondenterne yderst besværligt, da det ellers kunne have været med til at be- eller afkræfte deres påstande. En måde at omgå det på har været at forsøge at få interviews med flere personer i de valgte projekter, og dernæst benytte metodetriangulering, hvor der kan findes dokumentation for påstande via sekundære kilder. Det samme har gjort sig gældende for artefakter, hvor artefakterne har været det observerbare.

7.3.2 Organisationsstruktur

Til identificering af projekternes organisationsstruktur er der taget udgangspunkt i (Garzarelli, Limam, & Bjørn, 2008). Denne artikel opstiller to idealtyper: Katedral og basar organisationen. Organisationerne anses ud fra et arbejdsdelingsperspektiv. Katedral organisationen er topstyret med en centraliseret organisation⁴⁸, hvor imod at basarorganisationen er styret nedefra og op med en decentral organisation⁴⁹.

Katedralorganisationen kendetegnes ved, at et given projekt bliver topstyret og de nødvendige ressourcer bliver indhentet til projektet på baggrund af deres fagkundskaber. Hver person får tildelt en opgave, som personen skal løse. Det er ledelsens opgave at planlægge og uddelegere opgaverne.

Basarorganisationen kendetegnes ved, at det er op til den enkelte selv at vælge, hvilken opgave personen vil påtage sig, frem for at finde specifikke individer til en pågældende opgave. Ved denne type organisation vil det være muligt at trække på en stor mængde viden⁵⁰ fordelt på mange individer. Dette gør det også muligt, at flere personer kan arbejde på det samme problem, men ende ud med forskellige løsningsforslag.

⁴⁷ (Schultz, 1990) s. 32

⁴⁸ (Garzarelli, Limam, & Bjørn, 2008) s. 122

⁴⁹ (Garzarelli, Limam, & Bjørn, 2008) s. 122

⁵⁰ (Garzarelli, Limam, & Bjørn, 2008) s. 126

Afhængig af magtstrukturen i projektet kan det være et demokratisk valg. Bestemte individer, eller en kombination, der vælger en konkret løsning, der skal implementeres i projektet.

8 OSS projektanalyse rammeværk

Rammeværket er opbygget af fire faser, som er blevet belyst for hvert enkelt projekt. Før analysen er påbegyndt, er der udvalgt en række relevante projekter, som på hver sin måde har bidraget til en større forståelse af hvilke kritiske valg, der er opstået i løbet af et open source projekts livscyklus. Det er antaget i tesen, at de kritiske valgkonfigurationer vil bestå af fælles delmængder, som vil kunne sammenlignes på tværs af projekterne. Kriterierne for udvælgelsen har været præget af at finde projekter, som er startet på forskellige vis, og har været på forskellige stadier i deres livscyklus. Ved at fokusere på projekternes forskellighed har det været målet at få identificeret så mange forskellige kritiske valg som muligt, og derigennem kunne optegne de fælles delmængder.

De fire faser består af:

- Fase 1. Organisationsstruktur
- Fase 2. Kultur
- Fase 3. CBPP processen
- Fase 4. Kritiske valg

8.1 Organisationsstruktur

Første fase har taget afsæt i, hvorledes det enkelte projekt har været sammensat. Fokus vil være på organisationsstrukturen. Karakteristikken af projektet har taget udgangspunkt i basarorganisationen (Garzarelli, Limam, & Bjørn, 2008). Ved at bruge denne idealtipe som måleinstrument har det givet en dybere indsigt i, hvorvidt et projekt har haft mulighed for at drage nytte af de fordele, der har kunnet opnås igennem CPP processen. "*Unlike hierarchical and market modes of production, CBPP "relies on decentralized information gathering and exchange to reduce the uncertainty of participants" (Benkler 2002, p. 375), and is characterized by (i) decentralization of action and decision-making and (ii) utilization of social cues instead of prices or managerial authority as a means of motivating and organizing efforts (Benkler and Nissenbaum 2006).*" citat fra (Andreev, Feller, Finnegan, & Moretz, 2010, s. 3).

8.2 Kultur

Anden fase vil indeholde en kulturanalyse af det enkelte projekt baseret på fortolkningen af Scheins funktionalistiske kulturanalyse (Schultz, 1990). Kulturanalysen er valgt for at få en bredere forståelse af, hvorfor de involverede personer har ageret, som de har gjort. Ydermere kan kulturen i et projekt direkte modar-

bejde nogle af de fordele, som er skitseret i (Andreev, Feller, Finnegan, & Moretz, 2010). Denne artikel nævner blandt andet, at kulturen er en essentiel del af at kunne overkomme de organisatoriske udfordringer via koordination. "*Overcoming this organizational challenge related to defining boundaries, objectives, and ideology of an OSS project requires mutual agreement within the community regarding macroculture norms and beliefs.*" citat fra (Andreev, Feller, Finnegan, & Moretz, 2010, s. 11).

8.3 CBPP processen

Tredje fase består af en gennemgang af de seks punkter, som (Andreev, Feller, Finnegan, & Moretz, 2010) beskriver i deres procesdiagram⁵¹. Dette har sat de to første faser i relation til CBPP processen, hvori motivation også er blevet behandlet.

8.4 Kritiske valg

Femte fase har bygget videre på de tre foregående faser. Ved at kombinere organisationsstruktur, kultur og de seks punkter i CBPP processen herunder motivation med det teknologiske modenheds aspekt vil det være muligt at kunne identificere og forklare implikationerne ved det pågældende kritiske valg. Denne fase af analysen vil muliggøre sammenligning af de forskellige konfigurationer af kritiske valg.

Til måling af, hvor kritisk et valg har været, er der opsat følgende kriterier:

Kritisk	Beskrivelse	Implikationer
1	Et valg, som har haft en betydning for projektet.	Valget har understøttet eller moderarbejdet et andet valg.
2	Et betydningsfuldt valg	Valget har haft en væsentlig indflydelse, som har understøttet eller moderarbejdet et andet valg.
3	Et vigtigt valg	Valget har haft en betydning i forhold til struktur, kultur, motivation eller CBPP processen.
4	Et kritisk valg ⁵²	Valget har haft en væsentlig påvirkning på et eller flere forhold i forbindelse med struktur, kultur, motivation eller CBPP processen.
5	Et essentielt valg	Valget har haft en afgørende virkning på projektet, hvor flere forhold er blevet påvirket i forhold til struktur, kultur, motivation og CBPP processen.

Tabel 1 Definition af kritiske valg

⁵¹ Se Figur 2 Proces diagram "*flow of peer production process*"

⁵² Denne beskrivelse benyttes kun som et beskrivende element i forhold til, hvor kritisk valget har været. Et kritisk valg i alle andre henseende vil være i forhold til en skala på 1-5, som illustreres i Tabel 1.

9 Empiri

I dette multiple casestudie⁵³ er det valgt at benytte kvalitative stimulidata⁵⁴ til de primære data. Mere specifikt er der udført en række semistrukturerede interviews med nøglepersoner i de valgte OSS projekter. Denne type data er valgt på baggrund af ønsket om at opnå kendskab til, hvorledes de observerbare fænomener, kritiske valg, er konstituerede, samt hvilken effekt de har haft på det enkelte projekt. Da det på forhånd ikke har været muligt at forudsige, hvilke kritiske valg det enkelte projekt har stået overfor, eller hvilken virkning det har haft på projektet, har en kvalitativ fremgangsmåde givet en dybde som en kvantitativ fremgangsmåde ikke har kunnet. Det semistrukturerede interview er valgt, da det med belæg i teori og forfatterens egne erfaring, har været muligt at identificere mulige kritiske valg, samt hvad der har været med til at konstituere disse fænomener.

Der er så vidt muligt også benyttet sekundære data, som supplement til førnævnte interviews. De sekundære data har taget afsæt i åbne fora, dokumenter og VCS systemer, der har været tilgængelige for hvert projekt.

Afhandlingen har taget afsæt i tidligere situationer såvel som aktuelle situationer. Det optimale ville være at finde projekter, som stod ved et kritisk valg her og nu, men dette er set som værende usandsynligt. Det er derfor valgt at inddrage tidligere situationer, som har kunnet være med til at afdække mulige kritiske valg. At tidligere situationer er inddraget, som muligvis er flere år tilbage i tiden, har gjort, at reliabiliteten for de gældende situationer har været lav. Dermed også sagt at tidligere situationer har haft en væsentligt lavere værdi end udtalelser, som er kommet i forbindelse med aktuelle situationer.

9.1 Valg af projekter

Der er valgt en holistisk model til de forskellige projekter. Modellen bygger på fire spørgsmål og en række kriterier, hvor mindst ét kriterium per spørgsmål skal opfyldes:

Udvælgelsesspørgsmål	Kriterier
Hvordan startede projektet?	• Startede i kommerciel kontekst. • Startede i ikke-kommerciel kontekst.
Hvor befinder projektet sig nu?	• Afledt virksomhed. • Projektet oplever stor passivitet fra kerneudviklernes side.

⁵³ (Andersen, 2005) s. 113

⁵⁴ (Andersen, 2005) s. 151

Hvor succesfuldt har projektet været?

- Har mindst opnået en vis form for succes.

Hvor befinder kerneudviklerne sig?

- Mindst én kerneudvikler befinder sig i Danmark.

Målet med udvælgelsen har været at skaffe adgang til projekter, som hver især er startet på et forskelligt grundlag, og som har befundet sig forskellige steder i deres livscyklus.

9.1.1 Bromine

Udvælgelsesspørgsmål	Kriterier
Hvordan startede projektet?	Som et internt projekt hos Eniro Danmark A/S
Hvor befinder projektet sig nu?	Kernegruppen valgte den 7. november 2011 at nedlægge projektet med mulig overtalelse af andre.
Hvor succesfuldt har projektet været?	Der blev kørt mere end i gennemsnit 100.000 test om måneden ud af dem, som tillod anonym dataindsamling samt mere end 2000 unikke besøgende pr. måned. Samt et dengang aktivt forum med +600 brugere.
Hvor befinder kerneudviklerne sig?	Alle i Danmark.

Projektet er dels blevet udvalgt på baggrund af forfatterens involvering i projektet. Dette har betydet, at forfatteren har haft adgang til projektet gennem flere år, og dette er blevet benyttet i analysen. Projektet har desuden opfulgt de opstillede kriterier, og set i forhold til de to andre udvalgte projekter har det givet et projekt, som har haft nogle fælles træk, men også har adskilt sig fra de to andre projekter. Inden udarbejdelsen af denne afhandling er blevet påbegyndt, har det været kendt, at projektet snart ville stå overfor et kritisk valg. Det kritiske valg omhandlede om projektet skulle nedlægges, eller om der først skulle udgives en endelig version, som ikke var i beta, inden kernegruppen ville vælge at stoppe. Dette medførte en situation, hvor gruppen befandt sig i beslutningsprocessen i forhold til det kritiske valg.

9.1.2 Varnish Cache

Udvælgelsesspørgsmål	Kriterier
Hvordan startede projektet?	I forbindelse med en norsk avis, Verdens Gang og konsulentvirksomheden Linpro A/S ⁵⁵
Hvor befinder projektet sig nu?	Har support virksomhed, fx træning og support.

⁵⁵ Se Bilag 3

Hvor succesfuldt har projektet været?	Bliver bl.a. brugt af Facebook ⁵⁶ og Verdens Gang ⁵⁷ .
Hvor befinder kerneudviklerne sig?	Mindst en i Danmark.

Dette projekt er udvalgt, fordi det ligesom Bromine er udviklet i en kommerciel kontekst, og efterfølgende har afledt kommerciel support virksomhed. Det har vist sig senere, at udvikleren, som har stået bag størstedelen af projektet, ikke har været en del af den kommercielle afledte virksomhed, hvilket har resulteret i en perifer inddragelse af den kommercielle virksomhed i denne afhandling.

9.1.3 Exherbo

Udvælgelsesspørgsmål	Kriterier
Hvordan startede projektet?	Startede af én person, Bryan Østergaard.
Hvor befinder projektet sig nu?	For nyligt begyndt at have en stabil version.
Hvor succesfuldt har projektet været?	Projekts succes kan karakteriseres i forhold til, hvor mange udviklere der var samt mængden af kontributioner, der blev afgivet til projektet. På dette grundlag kan projektet anses som værende på vej til succes.
Hvor befinder kerneudviklerne sig?	Mindst en i Danmark.

Dette projekt er udvalgt som kontrast til de to forrige projekter. Dette projekt har ikke været indblandet i en kommerciel kontekst. Projektet har adskilt sig væsentligt i forhold til størrelsen på projektet, hvor de to forrige var drevet af nogle få individer, var Exherbo drevet af 19 kerneudviklere samt en projektejer. Community delen har også adskilt sig, hvor Exherbo-projektet opnåede at få en stor tilslutning af perifere udviklere, hvor de to forrige havde en væsentligt mindre succes med at få perifere udviklere til at bidrage til projektet.

9.2 Interviewguide

Interviewguiden består af tre hovedkategorier, henholdsvis spørgsmål vedrørende opstart af projektet, organisationsstruktur og kritiske valg.

Til at begynde med har interviewguiden til formål at danne en forståelsesramme for, hvordan projektet blev startet op, og hvilke vilkår der var gældende på daværende tidspunkt. Dette har givet en mulighed for

⁵⁶ Se Bilag 4

⁵⁷ Se Bilag 5

at få en indsigt i den transformation, som projektet har gennemgået. De overordnede spørgsmål er skitseret i Bilag 15 med ID A.X. Tabellen er bestående af en række spørgsmål og deres relation til teorien. Fx A.1. Hvordan startede projektet? Dette spørgsmål er relateret til, hvem omverdenen er, samt hvordan projektgruppen er placeret. Yderligere er det relateret til missionen, hvem vi er, og hvem vi er til for. Når spørgsmålene med ID A.X, er besvaret, fortsættes med spørgsmålene markeret med B.X. Disse spørgsmål har omhandlet organisationsstrukturen i projektet. Spørgsmålene er blevet gennemgået i flere omgange, da det har taget udgangspunkt i, hvordan projektet har været organiseret ved opstart, og hvordan projektet er organiseret i dag. Hvis projektet har været organiseret på andre måder end allerede afdækket, har en udbygning omkring omstændighederne været påkrævet. Sidste del af interviewet består af spørgsmålene C.X. Disse spørgsmål har været målrettet situationer, som respondenterne har betegnet havde haft en væsentlig betydning for projektet. I første omgang har det været muligt for respondenterne selv at nævne nogle situationer inden for de sidste tre måneder, som personen har karakteriseret, har haft væsentlig betydning for projektet. Der er set på de sidste tre måneder, da erindringen om situationen har været forholdsvis ny, og dermed ville den formentlig have en højere reliabilitet end situationer før tre måneder. Da der så vidt muligt er taget udgangspunkt i hele projektets levetid, har der efterfølgende været givet respondenterne mulighed for at nævne ældre situationer, som personen har betegnet som værende af væsentlig betydning for projektet. Afslutningsvis er respondenterne blevet udspurgt omkring de kritiske valg, der er blevet udformet i forhold til teori og forfatterens erfaringer. Fælles for alle spørgsmålene under C.X. har været, at de har relateret sig til flere områder; motivation, kultur, struktur.

Ud over interviewguiden er der udarbejdet et mindmap⁵⁸ som dirigentværktøj til interviewguiden. Dette mindmap har hjulpet med at komme med mulige udfald fra interviewene, hvorved der på forhånd har været forventninger omkring, hvilke mulige svar respondenterne ville komme med.

Til vurdering af, hvor kritisk et valg har været, blev det valgt initialt at lade respondenterne give deres bud på dette. Dette svar skulle kombineres med forfatterens vurdering af valget, men har vist sig at være mindre brugbart, da respondenterne har haft svært ved at give et definitivt svar på, hvor kritisk et valg har været. Derfor har forfatteren valgt at vurdere, hvor kritisk et valg har været ud fra de historier, respondenterne har fortalt. Ved at give respondenterne mulighed for at fortælle om en situation, har det været muligt på en mere neutral måde at udlede, hvor kritisk et valg har været. Det har givet anledning til højere reliabilitet, end hvis respondenterne havde svaret på, hvor kritisk et valg havde været på en foruddefineret skala.

⁵⁸ Se mindmap i Bilag 16

9.3 Interviewoversigt

Når personer er omtalt, er de blevet anført med deres forkortelse⁵⁹. Det er valgt, at alle interviews, på nær et skulle foregå ved personlig fremmøde. Et personligt fremmøde er valgt, da det har kunnet skabe en bedre personlig kontakt, end hvis det havde været via telefon eller chat.

Til udvælgelse af respondenter har kriteriet været, at de har skullet være en del af kernegruppen. I Bromine projektet har det været muligt at tale med alle kerneudviklerne, som har været en del af projektet igennem hele projektets livscyklus. Yderligere har den oprindelige testmanager fået stillet en række spørgsmål via e-mail for at indhente data omkring de overvejelser, der har været i forbindelse med opstarten af projektet. Det er ikke vurderet at være nødvendigt med et dybdegående interview, hvorfor e-mail interview er blevet valgt.

I Varnish projektet har det vist sig, at projektet reelt har været mindre end først antaget, da størstedelen af projektet er blevet drevet af PHK. For at få et bredere indblik i projektet er APJ også blevet udvalgt på baggrund af hans frivillige engagement i projektet.

Oprindeligt var det planlagt at inddrage et andet projekt end Exherbo, men da det viste sig ikke at være muligt at få adgang til projektet, blev Exherbo udvalgt. Det har siden vist sig, at valget af Exherbo har været med til at give en dybere indsigt i, hvilke typer OSS projekter der findes. Hvis det havde været muligt at få adgang til det oprindelige projekt, ville det formenligt have resulteret i et mindre alsidigt billede af OSS projekter. Da Exherbo forholdsvis sent er blevet inddraget i denne afhandling, har det påvirket dataindsamlingen, ved at kun én person er blevet interviewet. Via dialog med BØ blev det konstateret, at det ikke ville have været praktisk at interviewe andre, da dette ville have betydet rejser, som ikke ville have været en mulighed.

9.4 Sekundære kilder

De sekundære kilder er så vidt muligt blevet brugt til at understøtte påstande og historier, som er blevet frembragt under interviewene. Sekundære kilder, der er blevet brugt, er fora, mailingliste, projekternes hjemmeside og personlige blogs.

Sekundære kilder er desuden blevet brugt til at udmåle et communitys størrelse. Til denne udmåling er VCS og fejlrapporteringssystemer blevet brugt. Via dialog med respondenterne har det været gjort klart, at det interessante i forhold til et community har været den aktive del. Det billede, der er blevet skabt ved at benytte disse to kilder, har givet et objektivt billede af, hvor stort et community har været på undersøgelses-

⁵⁹ Se Bilag 24 for en interviewoversigt

tidspunktet. Dette har betydet, at det har været et meget konservativt billede, der er blevet tegnet, og det har skitseret et absolut minimum af, hvor stort et community har været. Da der ville forekomme dubletter ved at kombinere disse to kilder, er communityet blevet vist på to niveauer, hvoraf den højeste måling er blevet brugt, når der er talt om communityets størrelse.

9.5 Validitet

For at sikre en høj grad af validitet er semistrukturerede interviews valgt. Dette har betydet, at der ved at anvende en interviewguide har været fokus på de identificerede problemområder. Samtidigt har det også gjort plads til, at respondenterne har kunnet komme med deres perspektiv for at give et mere mangfoldigt billede af omstændighederne ved en given situation.

I de indsamlede data har der ikke været tegn på, at validiteten er blevet påvirket negativt, hvorved validiteten er betegnet for værende høj.

9.6 Reliabilitet

At tidligere situationer er inddraget, som muligvis har været flere år tilbage i tiden, har gjort, at reliabiliteten for de pågældende situationer har været lavere, end hvis de involveret havde stået i situationen på undersøgelsestidspunktet. Dermed sagt, at tidligere situationer vil have en væsentligt lavere værdi end udtalelser, som er kommet i forbindelse med aktuelle situationer. For at modvirke dette er flere personer blevet interviewet med udgangspunkt i samme interviewguide. Dette har resulteret i, at det har været muligt at sammenligne data indenfor det enkelte projekt, og derved øge muligheden for at finde anormaliteter, som ellers ville have påvirket graden af reliabiliteten negativt. Dette er gjort for Bromine og Varnish men ikke for Exherbo. For Exherbo har det betydet, at reliabiliteten har været lavere end de to andre projekter. For at modvirke dette er sekundære kilder blevet inddraget for alle tre projekter. Ved at benytte et semistruktureret valg har det endvidere været en mulighed at opdage fortolkningsproblemer og derved indgå i en mere detaljeret dialog for at udrede eventuelle problemer. Der er dog et problem med reliabiliteten i forhold til Bromine, da forfatteren var personligt involveret i projektet, og det vil være svært for andre at gengive resultaterne for dette projekt.

10 Bromine

Bromine var et værktøj til styring af automatiske softwaretests, som var bygget op omkring et andet OSS projekt (Selenium). Projektet startede i medio 2007 i virksomheden Kraks Forlag A/S, som senere blev opkøbt af Eniro Danmark A/S (efterfølgende refereret til som Eniro). Projektet begyndte med en række simple applikationer, der kunne afvikle eller opsamle resultaterne fra de automatiske tests til senere at være en webbaseret portal med kravstyring, kørsel af test og rapportering, blot for at nævne nogle af softwarens

egenskaber. Valget af licens kom i forbindelse med Eniros overtagelse af Krak. For at sikre bevarelsen af softwaren valgte den daværende testmanager at udgive softwaren som open source. I denne beslutning blev både testerne (testerne var også udviklerne af Bromine og vil fremadrettet blive refererede til som udviklere) og IT-direktøren inddraget. Resten af organisationen blev bevidst ikke inddraget for at forhindre eventuelle modstande, man kunne møde i den kommende konsoliderede virksomhed.

I projektet er der identificeret to skelsættende perioder. Den første periode var under opstarten af projektet omkring medio 2007 og ca. tre år frem, hvor den primære udvikling foregik på Eniro. Anden periode startede, da de tre kerneudviklere valgte at stoppe på Eniro og frem til at projektet afsluttedes ultimo 2011.

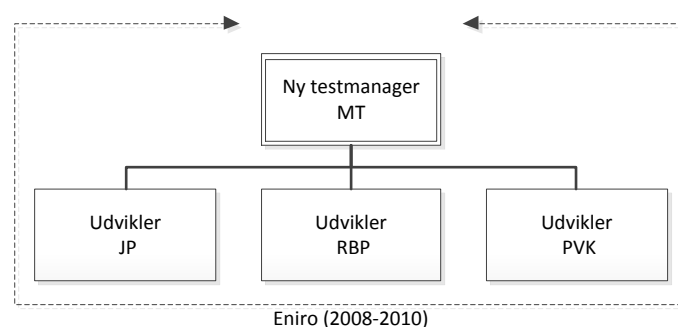
10.1 Organisationsstruktur

Bromine projektets organisationsstruktur var frem til 2010 præget af, at det primære arbejde blev udført i forbindelse med Eniro. Dette kom til udtryk ved, at der var en række interessenter i Eniro, som gerne ville have projektet i en bestemt retning. Strukturen omkring projektet var kendetegnet ved en fladstruktur, hvor det var testmanageren, som havde det overordnede ansvar. Under testmanageren var udviklerne, som alle havde lige meget autoritet.

10.1.1 Eniro 2007-2010

Projektet begyndte oprindeligt med fem personer (en testmanager og fire udviklere). I begyndelsen af 2008 var projektet reduceret til tre udviklere, og der var blevet ansat en ny testmanager. Transitionen foregik gnidningsfrit.

Videreudviklingen af Bromine var droslet ned i forbindelse med strammere budgetter og modningen af projektet. Det vil sige, at der derefter var tale om et stykke software, som leverede et ønsket resultat og var relativt stabilt. I denne periode benyttede udviklerne samtidig deres fritid på projektet.



Figur 4 Bromine organisationsdiagram⁶⁰ 2008-2010

⁶⁰ Test manager MT er Mette Tønder

10.1.2 Bromine 2010-2011

I 2010 valgte alle tre udviklere at fratræde deres stilling hos Eniro mod ansættelse i en anden virksomhed. I denne periode betød det også, at de ikke blev økonomisk støttet fra Eniro. Dette skift betød yderligere, at testmanageren, som tidligere havde ansvaret, ikke længere var en del af projektet. Efter deres opsigelse blev udviklingen primært rettet mod support og for at rette fejl i softwaren. I november 2011 meddelte udviklerne officielt, at de ikke fremadrettet ville videreudvikle eller supportere på Bromine. I forbindelse med deres udmeldinger opfordrede de andre til at overtage projektet, hvilket resulterede i nogle få henvendelser.

Ledelsesmæssigt betød det for projektet, at de tre udviklere blev anset med den samme mængde autoritet som tidligere, men manglen på en person, som tog ansvar for projektets overordnende mål, fik konsekvenser for projektet.



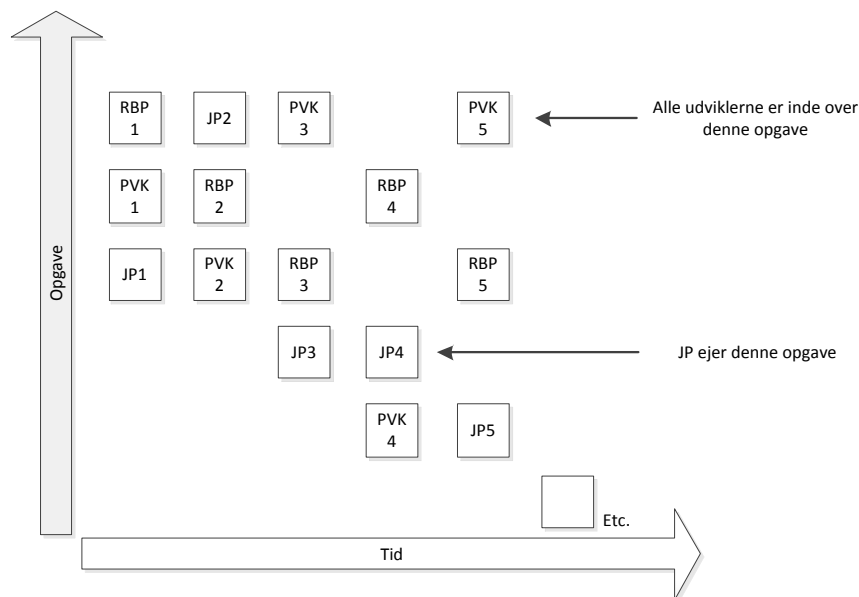
Figur 5 Bromine organisationsdiagram 2010-2011

10.1.3 Arbejdsdeling

Alle tre perioder lignede meget hinanden set på arbejdsdelingen. I forbindelse med arbejdet udført for Eniro blev der fastsat nogle forholdsvis frie rammer for udviklerne. Det var testmanageren, som opsatte disse rammer, og det var dernæst op til den enkelte udvikler at vælge, hvilke dele han ville arbejde med.

Set i forhold til de to idealtyper hos (Garzarelli, Limam, & Bjørn, 2008) er denne betegnet som en hybrid organisation. Arbejdsdelingen er skitseret i Figur 6.

De opgaver, udviklerne valgte at påtage sig, var som udgangspunkt ikke nogen, som den enkelte udvikler havde ejerskab over, men i visse tilfælde resulterede det i, at en person endte som den eneste, der arbejdede på en given opgave.



Figur 6 Visuel repræsentation af arbejdsdelingen i Bromine⁶¹

Efter bruddet med Eniro var det op til udviklerne at fastsætte rammerne for retningen på projektet. Initialt betød det ikke noget, da udviklerne stadig selv brugte softwaren, og derfor varetog deres egne interesser i forhold til deres anvendelse af softwaren. Ca. et halv år efter bruddet med Eniro brugte udviklerne ikke længere softwaren. Dette fik betydning for projektet, da deres eneste input til, hvilken retning projekt skulle gå, kom fra deres forestilling omkring, hvordan udviklerne troede, at softwaren blev anvendt af brugerbasen. Det blev forsøgt at inddrage brugerne men uden held.

10.2 Kultur

I forbindelse med de to skelsættende perioder er der kun udarbejdet én kulturanalyse. Det kunne være nødvendigt med særskilte analyser for hver periode, men idet der ikke var væsentlige ændringer i kulturen fra start til slut, er der i stedet fremhævet differentieringerne fra hver periode sideløbende med analysen. Ydermere har analysen kun taget udgangspunkt i udviklingsteamet af Bromine. Det har medført, at analysen kun vil omhandle den gruppering bestående af udviklere og testmanager direkte relateret til projektet i den tid, hvor udviklingen også foregik på Eniro. Resten af organisationen er fravalgt, da denne ikke har haft direkte indflydelse på projektet.

⁶¹ Dette er en simplificering af arbejdsdelingen, for at illustrativt at visse hvordan organisationen ser ud i forhold til katedral og basar organisationen.

10.2.1 Artefakter

10.2.1.1 Fysiske udtryk

Tre arbejdspladser

I forbindelse med Bromine fandtes tre arbejdspladser. Arbejdspladserne var hhv. på Eniro, som var et stort åbent kontorlandskab, i et sommerhus, som var en fast tradition i en årrække og hjemme hos den enkelte udvikler.

Åbent kontorlandskab

Det åbne kontorlandskab hos Eniro var med til at gøre kommunikationsvejene mindre mellem ansat og leder, da den fysiske afstand var indenfor et par meter.

Uformel påklædning

I testteamet på Eniro var påklædningen uformel. Dette gjaldt både testmanager og udviklere. Hvis man så på resten af organisationen, ville man kunne finde bestemte påklædningsmønstre, fx gik seniorledelsen i habitter, men som nævnt vil resten af organisationen ikke blive behandlet. Det, som adskilte medarbejderne, var deres titler, fx testmanager, CEO osv. I OSS delen var der titler som moderator og kerneudvikler.

OpenQA

I forbindelse med optagelsen hos OpenQA⁶² fik Bromine adgang til en webportal, hvor de kunne fortælle omverdenen om projektet. På denne portal blev deres mission artikuleret i form af "*Its lofty aim is to be the open source alternative to commercial tools (such as HP Quality Center).*". Dette var missionen, som blev fastlagt i og omkring tidspunktet, hvor Bromine blev optaget hos OpenQA. Senere i projektet var denne mission blot en notits i "README"⁶³ filen.

10.2.1.2 Sprog

Uformel tale

Sproget både i og uden for Eniro var meget uformelt, og det var med til at gøre omgangstonen mere personlig. Blandt de tre udviklere var sproget mere ungdommeligt og bar præg af tidens udtryk set i forhold til lederens sprog, som var mere afdæmpet og modent. Engelske udtryk blev brugt naturligt, da der blev talt om tekniske problemer og løsninger i forbindelse med programmering. Udviklerne arbejdede meget autonomt, men da der var brug for afklaring eller godkendelse til at arbejde på nye områder, var det lederen, der bliver spurgt til råds.

⁶² OpenQA var et samlingssted for open source med relevans til Selenium. OpenQA skiftede senere navn til SeleniumHQ. For mere information se <http://seleniumhq.org>

⁶³ Se Bilag 6

Forum

Efter optagelse hos SeleniumHQ blev der stillet et forum til rådighed, som blev udviklernes adgang til kommentar, feedback eller kritik fra communityet, som dermed spændte udover Eniros grænser. Tonen på dette skriftlige medie var holdt formelt for at skabe et seriøst og troværdigt billede af produktet.

"Det skal bare lige kunne"

I forbindelse med udviklingen forekom der udtryk som "det skal bare lige kunne". Det var en indikation på, at målet med projektet ikke var fast defineret. Features og ændringer blev til, da behovet opstod eller man først havde afprøvet en løsning men ikke var tilfreds med resultatet. Dette afspejlede også udviklernes mangel på erfaring i starten af projektet.

Fagsprog

Sproget i teamet var præget af, at de arbejdede inden for softwaretestbranchen. Hvilket betød, at der forekom fagudtryk som test cases og requirements. Denne terminologi dannede også grundlag for flere mål, da betydningen af en række udtryk blev set som centrale dele af, hvad produktet skulle kunne.

10.2.1.3 Teknologi

CakePHP

I et forsøg på at forbedre arkitekturen valgte udviklerne at omskrive hele kodebasen baseret på et udviklingsrammeverk, CakePHP⁶⁴. Denne nye version blev tildelt version 3.

Wiki

Til dokumentation og vejledninger fandtes en wiki-side, som var frit tilgængelig på projektets hjemmeside. Dokumentation som fandtes var mangelfuld, og flere emner var ikke længere gyldig på grund af ændringer i softwaren.

Distributioner

Til distributionen af softwaren benyttede projektet flere kanaler. Der var SVN, som var det revisionskontrollsystem, som projektet benyttede, en zip fil, som indeholdte den seneste udgivelse og et virtual machine billede⁶⁵, hvor en Linux server var installeret med Bromine. Tidligere havde projektet også en self-installer⁶⁶.

⁶⁴ For mere information omkring CakePHP gå til <http://cakephp.org/>

⁶⁵ Også kaldet et virtual appliance

⁶⁶ Self-installer er et installationsmedie, som kan installere et program automatisk. Slutbrugeren skal derfor ikke bekymre sig om opsætning af hverken softwaren eller 3. parts programmer.

10.2.1.4 Tradition

DIY prototyping

I projektet var der en tradition for, at den person, der fik en ide, også var den person, som implementerede den. Det var som hovedregel kun, da der var tale om større tiltag, at de andre udviklere var involverede i selve implementeringen. Dog blev der ofte talt og diskuteret om ideerne i plenum. Udviklingsmodellen kunne betegnes som prototyping.

Autonom ledelse

Ledelsesmæssigt fik udviklerne lov til selv at bestemme en stor del af, hvad der skulle udvikles, når det var i forbindelse med arbejdet på Eniro. Lederens rolle i forbindelse med projektet var at få organisationens interesser og behov dækket. Da disse behov begyndte at blive dækket tilstrækkeligt i form af produktets kunnen, blev der også skåret ned på den tid, Eniro ville investere i projektet. Ledelsesstilen kunne betegnes som konsensusføgende demokrati, hvor alle var lige. Ved tvivl blev testmanageren spurgt til råds.

Sommerhustur

To gange i 2009-2010 var der en tradition for at udviklerne tog i sommerhus og brugte en uge på at videreudvikle produktet. Det var ikke unormalt, at de brugte 14 timer om dagen på udviklingen i denne ene uge om året.

Forældet dokumentation

I projektet var der ikke tradition for at skrive dokumentation. Den dokumentation, som blev lavet, endte ofte med at være forældet, da udviklingen hurtigt skiftede retning på projektet.

10.2.1.5 Historier

Historier i dette projekt var ikke noget, der typisk blev omtalt, men der fandtes en række historier, som kom op til overfalden, såfremt samtalen faldt på projektets livscyklus.

CakePHP

En væsentlig historie var, da udviklerne valgte at skifte til CakePHP. Denne historie afspejlede udviklernes tekniske kundskabers udvikling fra at være nybegyndere til at kunne strukturere et større system og undgå faldgrupper. Skiftet betød en komplet genskrivning af hele kodebasen og et brud med tidligere versioner.

Forretningsmulighed

En anden historie udspillede sig i forbindelsen med et anerkendt firma, som udviklerne fik en aftale med, hvor de skulle udvikle en feature, som kunne integreres i deres service. Dette var det første forsøg på at kommercialisere en komplementerende ydelse i projektet. Da der taltes om monetære emner, faldt samta-

len desuden på en historie omkring en enkelt donation, projektet havde modtaget fra Japan. Denne enkelte donation, som var et mindre beløb, havde ikke nogen betydning for projektet som helhed, men værdien i handlingen havde stor betydning for udviklerne, og det var med til at styrke udviklernes tro på projektet.

Udviklerne sagde op på Eniro

Den sidste historie var i forbindelse med stoppet på Eniro, hvor alle tre udviklere valgte at sige deres stillinger op samtidigt. Konsekvensen var, at ingen overtog lederrollen for projektet, som var testmanagerens opgave.

10.2.2 Værdier

10.2.2.1 Mål

Fællesskab

Ved projektets begyndelse var et af testmanagerens mål, at projektet skulle være med til at fastholde medarbejderne, styrke sammenholdet samt forbedre kompetencerne i teamet. Dette var også en af grundene til, at projektet blev gjort open source.

Brugervenlighed

I projektet blev brugervenlighed anset som en væsentlig del af projektet. Det var derfor vigtigt for projektet, at det var nemt at komme i gang med, hvilket resulterede i flere distributionsformer, som blev benyttet på grund af denne holdning til brugervenlighed.

Sidste udgivelse

Kerneudviklerne havde den holdning, at de gerne ville afslutte projektet med en sidste udgivelse, da projektet var stagneret ved version 3 RC2⁶⁷, men det skete aldrig.

10.2.2.2 Midler

Pair development / testing

Pair development og testing blev introduceret som et middel til at opnå målsætningen om fællesskab. Dette var endvidere med til at styrke kompetencerne i teamet.

Læring som drivkraft

Læring som drivkraft var et vigtigt element, da nogle af de andre opgaver i forbindelse med arbejdet på Eniro, var præget af monotont arbejde. Et middel for at fastholde udviklerne, var at de fik en stor grad af frihed, når det kom til Bromine.

⁶⁷ Release candidate er en betaudgivelse, som potentielt kan gå hen og blive den endelige version, med mindre nogle kritiske fejl opdages. Se Bilag 18.

Minimalistisk udviklingsproces

I forbindelse med arbejdet på Bromine var processen præget af, at der ikke var fastlagt en proces for, hvordan projektet skulle styres. Dette resulterede i det, som kan beskrives som prototyping med et DIY tanke-sæt.

10.2.2.3 Måling

Versioner

I projektet fandtes der ikke deciderede målingsinstrumenter, men i stedet når der blev talt om hvor langt udviklingen var, blev der benyttet versionsnumre på de enkelte udgivelser. Der var ikke fastsat regler for, hvordan at de forskellige versioner skulle versioneres.

10.2.2.4 Korrektion

Konsensus søgende demokrati

Når der var behov for korrektion, var det i samråd med alle udviklerne og til tider også testmanageren, hvor der blev lagt vægt på et konsensusøgende demokrati. Hvis testmanageren, da udviklingen også foregik på Eniro, mente, at projektet havde behov for at blive korrigeret, havde testmanageren bemyndigelse til at trække projektet i den ønskede retning.

10.2.2.5 Gruppegrænser og kriterier for optagelse og udelukkelse

Bidrag og få mulighed for at blive en del af teamet

Inden at Bromine blev gjort tilgængeligt på SeleniumHQ, var den eneste mulighed for at deltage i projektet, at blive ansat i testafdelingen i Eniro. Ved at projektet blev optaget på SeleniumHQ, fik projektet adgang til nye potentielle ressourcer. Det var altid en mulighed at blive en del af teamet, men i begyndelsen var projektet præget af en høj indlæringskurve og en lukket kernegruppe. Senere i projektet faldt kompleksiteten og kerneudviklerne forsøgte at åbne op for kernegruppen.

10.2.2.6 Magt og status

Ansvar for egen kode

Magten fordeltes primært på baggrund af, hvem der fandt på en ny ide. Typisk var det personen selv, der implementerede ideen og vedligeholdte den. Alle havde dog mulighed for at arbejde på ideen.

De tre oprindelige udvikleres barn

Da det blev besluttet, at projektet skulle nedlægges og evt. videregives, skulle det ikke bare være til hvem som helst. I et nyhedsbrev offentliggjorde udviklerne, at hvis nogen var interesserede, så kunne udviklerne kontaktes og en aftale om videregivelse af projektet kunne finde sted. I månederne efter nyhedsbrevet udkom, kom der en håndfuld henvendelser, hvor alle blev spurgt om, hvem de var, om de havde de tekni-

ske kompetencer påkrævet for at videreførere projektet, samt hvilke tanker de havde gjort sig for projektets fremtid. Det var vigtigt for udviklerne, at projektet landede i de rigtige hænder.

10.2.2.7 Ledelse

Autonom ledelse

Den autonome ledelse blev anset som en vigtig del af projektet, da udviklerne derved følte en stærkere tilknytning til projektet.

10.2.2.8 Standarder

MVC

Inden tredje version af Bromine var der reelt ikke nogen standarder for, hvordan udviklingen skulle foregå. Arkitekturen blev løbende udformet, og der var ingen klare retningslinjer for, hvordan udviklerne skulle arbejde. Med Bromine 3 valgte udviklerne at smide al koden ud og starte forfra. Til valg af arkitektur bestemte udviklerne at benytte et rammeværk, CakePHP, som benyttede et model, view og controller (MVC) design. Ved at vælge dette rammeværk betød det, at der skulle overholdes en række kodestandarder, som CakePHP foreskrev, og derved tvang det udviklerne til at forholde sig til en ensformig og konsistent arkitektur.

10.2.3 Grundlæggende antagelser

"Alle er lige"

Der var en grundlæggende antagelse om at alle, som blev en del af kernegruppen, var lige. Når der skulle tages en beslutning, som havde større konsekvenser for projekter, blev beslutningen taget op i gruppen, hvor man diskutererede fordele og ulemper, og om man skulle forsætte i den retningen, som beslutningen pegede. Hvis der var tale om mindre beslutninger uden større konsekvenser, havde den enkelte udvikler selv mandat til at foretage beslutningen.

"Vi kan ikke tvinge andre til at deltage aktivt"

Den grundlæggende antagelse om, at man ikke kunne tvinge andre til at deltage, betød reelt, at der ikke blev gjort noget særlig for at rekruttere nye medlemmer til projektet. Det blev antaget, at hvis en person havde lyst til at deltage, kom personen til projektet. Personens engagement blev derved ikke håndteret udover at give personen adgang til projektets systemer.

"Hvis man ikke bruger softwaren, er projektet ikke interessant"

Denne grundlæggende antagelse antog, at den primære interesse i projektet var i forhold til udviklerens arbejde. Dette blev åbenlyst, da kerneudviklerne valgte at sige op på Eniro, og deres tidsforbrug på projektet faldt markant. Da det skulle besluttet, hvorvidt projektet skulle udgive en endelig version af Bromine 3,

faldt et af hovedargumenterne på, at udviklerne ikke selv ville bruge hverken en færdig version eller en ny modificeret udgave af Bromine.

"Vi har et ansvar overfor communityet"

I forlængelse af antagelsen om, at hvis man ikke brugte softwaren, så var projektet ikke interessant, antoges det i organisationen, at udviklerne havde et ansvar overfor communityet. Dels fordi de selv ville se godt ud overfor brugerne og dels fordi, man havde fået opbygget en rimelig størrelse community. Dette var den primære drivkraft, da kerneudviklerne valgte at stoppe på Eniro. Hvis projektet ikke havde haft denne grundlæggende antagelse, ville projektet sandsynligvist have været stoppet længe før, det gjorde.

"Projektet er godt for fællesskabet"

Under projektets konceptualisering var det testmanagerens målsætning, at projektet skulle være med til at styrke sammenholdet og skabe en god holdånd. Denne antagelse om, at projektet var godt for fællesskabet blev styrket gennem årene. Projektet blev omdrejningspunkt for en del af tiden i og uden for arbejdstiden.

10.2.4 Det kulturelle paradigme

Det kulturelle paradigme for Bromine, skitseret i Figur 7, er blevet centreret omkring antagelsen om at projektet er godt for fællesskabet. Denne antagelse var ikke den mest fremtrædende, men den forbandt resten af de grundlæggende antagelser, der blev identificeret i projektet. Ud fra denne antagelse udsprang både antagelsen om, at "alle er lige", og at projektmedlemmerne havde et ansvar overfor communityet. "Alle er lige" blev styrket gennem antagelsen om, at projektet var godt for fællesskabet, hvis der blev differentieret i, hvor meget magt den enkelte udvikler havde, ville det have haft en negativ effekt, fordi udviklerne alle kom fra lignende uddannelser, samt at deres erfaring var begrænset. Det betød, at ingen var i en bedre position til at tage en beslutning end andre. Desuden havde testmanageren det overordnede ansvar. Da alle blev anset for lige, foruden testmanageren, betød det, at ingen havde mandat til at tvinge andre til at udføre bestemte opgaver. Senere i projektet, da man forsøgte at inddrage andre uden for Eniro, betød det, at de nye ville få tildelt de samme vilkår. Dette havde den effekt, at de nye udviklere blot fik adgang til systemerne og reelt aldrig blev en del af projektet, da de kun var med i en kortere periode, selvom der var potentiale for et større engagement. Antagelsen om "vi kan ikke tvinge andre til at deltage aktivt" havde desuden en afledt effekt i tilfældet af, at en nytilkommen udvikler fortsat ville have det styrkede fællesskab. Samtidig kunne det udledes, at man også fik fraserteret folk, som ikke var tilstrækkeligt engagerede. Organisation var aldrig gearret til at fastholde nytilkommendes engagement, hvilket betød, at personer, som oprindeligt viste et stort engagement, langsomt mistede lysten til at deltage.

Antagelsen om, at udviklerne havde et ansvar overfor communityet udsprang fra, at projektet var godt for fællesskabet. Ved at projektmedlemmerne havde en tilknytning til projektet og en tro på, at det man gjorde, var rigtigt, blev fællesskabet yderligere styrket gennem kontakten til communityet, hvilket resulterede i, at projektmedlemmerne følte et ansvar overfor communityet. Kvaliteten skulle være til stede, og udviklerne skulle yde support på projektet, da både det passive og aktive community blev værdsat højt. Denne antagelse var desuden det største argument for, at projektet skulle færdiggøres med en endelig udgave af Bromine 3. Denne antagelse relateredes desuden til motivationen for at fortsætte projektet, hvor antagelsen "hvis man ikke bruger softwaren, er projektet ikke interessant", var modvægten til ansvaret overfor communityet. Da antagelsen om, at "projektet ikke var interessant, hvis man ikke brugte softwaren", blev dominerende, betød det, at det ville påvirke antagelsen om, at "projektet er godt for fællesskabet" negativt, hvilket i sidste instans betød, at projektet stoppede.



Figur 7 Bromine kulturelle paradigme

10.3 CBPP

Projektet var beskyttet gennem GNU GPL v.3. Denne licens sikrede, at man måtte bruge softwaren, som man havde lyst, at man måtte ændre softwaren, at man måtte dele softwaren samt sine ændringer (Smith). Disse betingelser kunne ikke fraviges og ville altid følge softwaren både i original og modificeret form. Denne licenstype kunne komplicere sammenspillet imellem den kommercielle verden og open source delen, da ændringer og samspillet mellem proprietæresystem kunne umuliggøres af den valgte licens. Projektet var derved beskyttet og tilgængeligt gennem de webbaserede teknologier, som projektet benyttede sig af, hvilket betød, at projektet havde alle forudsætningerne for at kunne drage nytte af ressource allokeringens fordel.

Det potentielle, der ligger i ressourceallokeringsfordelen, kan blive realiseret, når projektet udnytter de muligheder, der lægger i informationsbehandlingsfordelen, som munder ud i en bred brugerdeltagelse og filtrering af kontributionerne til projektet. I dette projekt var der en lille brugerdeltagelse set på kernegruppen, hvor der ikke var nogen fastdefinerede roller. Alle havde mulighed for at arbejde på de dele af systemet, som de selv fandt interessant. Der var dog opstået visse reservater, som den enkelte udvikler var ale ne om. Disse reservater opstod i forhold til DIY mentaliteten, hvor det implicit var givet, at den enkelte udvikler ville være den bedste til at varetage videreudviklingen på området, da person allerede besad en viden vedrørende implementeringen. Projektets arkitektur var i version 1 og 2 en begrænsning i forhold til at stimulere en bred brugerdeltagelse. Koden var præget af udviklernes mangel på erfaring, hvilket betød, at koden var svær at gå til, da der manglede en konsistent arkitektur, og det skabte en unødigt kompleksitet. I version 3 blev hele kodebasen omskrevet og bygget på et anerkendt rammeværk. Dette var med til at skabe en transparent arkitektur. Ydermere gav det anledning til et modulbaseret system, hvor det var muligt for andre at lave plugins til systemet. Dette blev aldrig en succes udadtil, da projektets kultur ikke var i stand til at bære nytilkommende til det punkt, hvor de følte et ansvar for projektet, og derved aldrig blev en del af kernegruppen. Selvom projektet kun bestod af en lille kernegruppe, var der kontakt til communityet, som var med til at give input til projektet. For at filtrere inputtene og kontributionerne blev de opbevaret i et fejlhåndteringssystem. Mange af de inputs og kontributioner, der blev indgivet til projektet, endte dog med aldrig at blive håndteret og strandede derefter i fejlhåndteringssystemet. Dette var givetvis også med til at begrænse en bredere brugerdeltagelse, da det havde en afskrækkende signalværdi.

Både ressourceallokerings- og informationsbehandlingsfordelen kræver, at man overkommer motivationsudfordringen. Projektet begyndte med at være drevet af ydre motivationsfaktorer, idet projektet begyndte i forbindelse med udviklernes arbejde. Senere i projektet skete et skift fra denne ydre motivationsfaktor til også at inkludere indre motivationsfaktorer. Dette kunne tilbageføres til den oprindelige testmanagers beslutning om at skabe et projekt, som havde en arbejdsmæssig funktion samt være med til at styrke sammenholdet. Ved at give udviklerne den frihed til selv at kunne bestemme retningen på projektet, var det muligt at stimulere den fællesskabsbaserede motivation, som gjorde, at udviklerne følte et ansvar overfor projektet og derigennem communityet. Læring og udfordring spillede også en rolle i udviklernes motivation, men det blev overskygget af den monetære godtgørelse og den fællesskabsbaserede motivation. Dette skyldtes, at læring og udfordringen ved at arbejde med et nyt område hovedsageligt var gældende, da arbejdet foregik på Eniro, hvor den monetære godtgørelse var den primære kilde til motivation. Senere i projektet spillede læring og udfordringen en mindre rolle, da udviklingen gik over til primært at bestå af vedligeholdelse. Skalerbar motivation blev en realitet i version 3, men grundet kulturen var det ikke muligt at drage fordel af denne type motivation.

Når ressourceallokerings- og informationsbehandlingsfordelene er håndteret, opstår der et behov for koordineret handling, som er en del af organisationsudfordringen. Som det er beskrevet i afsnit 10.1.3 om arbejdsdeling, kunne projektet betegnes som en hybrid organisationen, hvor koordinering på makroniveau blev foretaget af testmanageren, men på mikroniveauet var det op til den enkelte. Beslutninger, som lå imellem disse to niveauer, blev taget op demokratisk. Mange af beslutningerne blev diskuteret internt i projektet, og typisk var den personen, som kom med ideen, der implementerede den. Denne type organisering havde potentiale for at kunne udnytte både ressourceallokerings- og informationsbehandlingsfordelen, da den opsatte rammer for projektet, men det var op til den enkelte at finde områder, som personen fandt interessant. Problemet som nævnt tidligere lå i projektets kultur, som i realiteten modarbejdede nye udvikleres incitament for at deltage i projektet.

Efter at opgaverne i projektet er koordineret, opstår behovet for integration. Integrationen fungerede kun delvist på grund af projektets størrelse. Alle blev anset som lige og havde fuld skriveadgang til projektets CVS, hvilket betød, at løsninger lavet af den enkelte udvikler blev commitet direkte ind i systemet. Når der blev bidraget en patch fra den aktive del af communityet, endte den oftest i fejlhåndteringssystemet, hvor den kunne blive liggende længe før den eventuelt blev commitet. Dette betød, at integrationen fungerede i det omfang at nye ideer blev diskuteret internt i projektet, før de blev implementeret.

10.4 Kritiske valg

10.4.1 K1 - Licensvalg

For projektets senere overlevelse var det essentielt, at det blev udgivet under en OSS licens⁶⁸. Det var ikke en afgørende faktor, at valget faldt på GNU GPL v2⁶⁹. Da udviklerne stoppede på Eniro, var det deres forpligtigelse og fællesskabsbaseret motivation, som gjorde at projektet fortsatte i yderligere halvandet år. Hvis projektet var udgivet under en proprietær licens, ville projektet været et internt projekt i Eniro og ejerskabsfølelsen, som udviklerne havde over projektet, ville ikke have været til stede. Dette ville have betydet, at udviklerne ikke ville have brugt deres fritid på at udvikle softwaren i perioden frem til deres opsigelse, samt at det ikke havde været muligt at videreføre projektet efterfølgende.

10.4.2 K2 + K6 - Arkitektur

Frem til og med version 2 var arbejdet præget af udviklernes manglende erfaring. Koden var fx ustruktureret. Der var dog gjort forsøg på at bevare en overordnet arkitektur men uden klare retningslinjer eller kon-

⁶⁸ Der tages ikke højde for alle implikationer ved at softwaren blev udgivet som OSS, fx at det åbner op for et community, da dette anses som væren en normal del af et OSS projekt, og derved ikke kan betegnes som værende kritisk for projektet.

⁶⁹ Senere opdateret til v3.

ventioner for god kode. Dette betød, at der var en høj indlæringskurve for andre udviklere. Dette kan have været med til at begrænse andre udvikleres interesse i projektet. Skiftet til Bromine 3, hvor der blev benyttet et udviklingsrammeverk, CakePHP, afspejlede en modning af udviklerne. Skiftet til et anerkendt rammeverk betød at indlæringskurven ville være betydeligt lavere for nytilkommende, såfremt at personen havde et kendskab til rammeverket. Efter skiftet til CakePHP kom få perifere udviklere til. Der var dog ikke noget entydig bevis for, at et skift var med til at øge mulighederne for flere perifere udviklere eller kerneudviklere. En forklaring på dette kan være, hvilket markedssegment Bromine henvendte sig til. I forhold til passive brugere oplevede Bromine en stor supportbyrde, hvilket indikerede, at brugerne ikke havde den tekniske viden, der skulle til, da mange problemer kunne spores tilbage til fx opsætning af webservere og installering af tredjepartsprogrammer.

Set i forhold til CBPP processen var det den rigtige beslutning at skifte til CakePHP, da dette betød en ændring af arkitekturen, som gjorde softwaren modulær og granulær, hvilket var en af betingelserne for en bred brugerdeltagelse. Valget havde derfor en potentiel kritisk værdi på fire, men da en bred brugerdeltagelse aldrig blev realiseret, kan dette valg ikke anses for værende mere kritisk en to. Efter version tre var udgivet, skete der en udvikling i forhold til aktive brugere, der bidrog i form af fejlrapporter og featureanmodninger. Dette er anvist i Figur 8, hvor der ud for K6 er en stigning i personer, der har involveret sig, hvilket understøtter indikationen om, at de aktive brugere ikke var tekniske, da fåtallet indsendte patches, selvom folk blev opfordret til det⁷⁰.

10.4.3 K3 + K7 - Autonom ledelse

Da projektet begyndte i kommerciel henseende, var det essentielt, at udviklerne fik mulighed for at være autonome. Autonomiteten styrkede fællesskabet ved at de selv fik lov til at styre retningen på projektet. Ydermere påvirkede det udviklernes motivation i forhold til den fællesskabsbaserede motivation. Dette betød en styrkelse af ejerskabsfølelsen over projektet, som medvirkede til at projektet blev videreført efter deres opsigelse på Eniro. Problematikken for denne ledelsesform var, at i open source henseende manglede der en lederskikkelse. Der var kun opsat retningslinjer for projektet internt i Eniro. Dette problem blev cementeret, da udviklerne forlod Eniro, og ingen overtog rollen som leder i projektet. Uden leder i projektet resulterede det i, at udviklerne ikke skulle stå til regnskab over for nogen. Kombineret med at den primære drivkraft var baseret på monetær godtgørelse, resulterede det i et stagnerende projekt. Mangel på lederskab i open source delen, kunne relateres til manglen på evnen til at fastholde nytilkommende udviklere. I forhold til CBPP processen var det mangel på koordineret handling, som fastholdte den lave

⁷⁰ Se Bilag 17 for eksempel.

perifere udviklerdeltagelse. Ledelsesformen kunne endvidere henføres til projektets kultur, som reelt havde frastødt ideen om lederfigurer.

K3 er derfor betegnet som værende et essentielt valg for projektet og K7 som værende en afledt effekt af K3 med en vis betydning for projektet.

10.4.4 K4 - OpenQA

Ved at blive optaget ved OpenQA var det med til at opfylde første kriterier i CBPP processen, at projektet skulle være tilgængeligt. Uden OpenQA ville det stadig have været muligt at opfylde dette kriterium, men den eksponering, projektet fik via OpenQA, ville have været nær umulig at generere, hvis udviklerne havde gjort det på egen hånd. Eksponeringen gav projektet adgang til en potentiel bred brugerdeltagelse. Dette valg er derfor betegnet som værende essentielt for projektet, da denne samarbejdspartner var med til at opfylde en række kriterier i CBPP processen.

10.4.5 K5 + K8 - Distributioner

Gennem projektets livscyklus blev der gjort flere forsøg på at imødekomme brugernes tekniske niveau eller mangel på samme. Første forsøg var at udgive en self-installer, præcis hvor meget gennemslagskraft den havde fandtes der ingen optegnelser over, men det var begyndelsen på, at softwaren skulle være så nem at anvende som muligt. Senere i projektet blev softwaren udgivet som et virtual appliance, hvilket resulterede i, at når brugerne kom med problemer vedrørende opsætning af tredjepartsprogrammer, som Bromine var afhængig af, blev de adviseret om at downloade det virtual appliance i stedet. Dette resulterede i en betydelig reduktion af supportspørgsmål.

Dette var primært med til at fjerne en negativ effekt på udviklernes motivation, da supportbyrden blev reduceret.

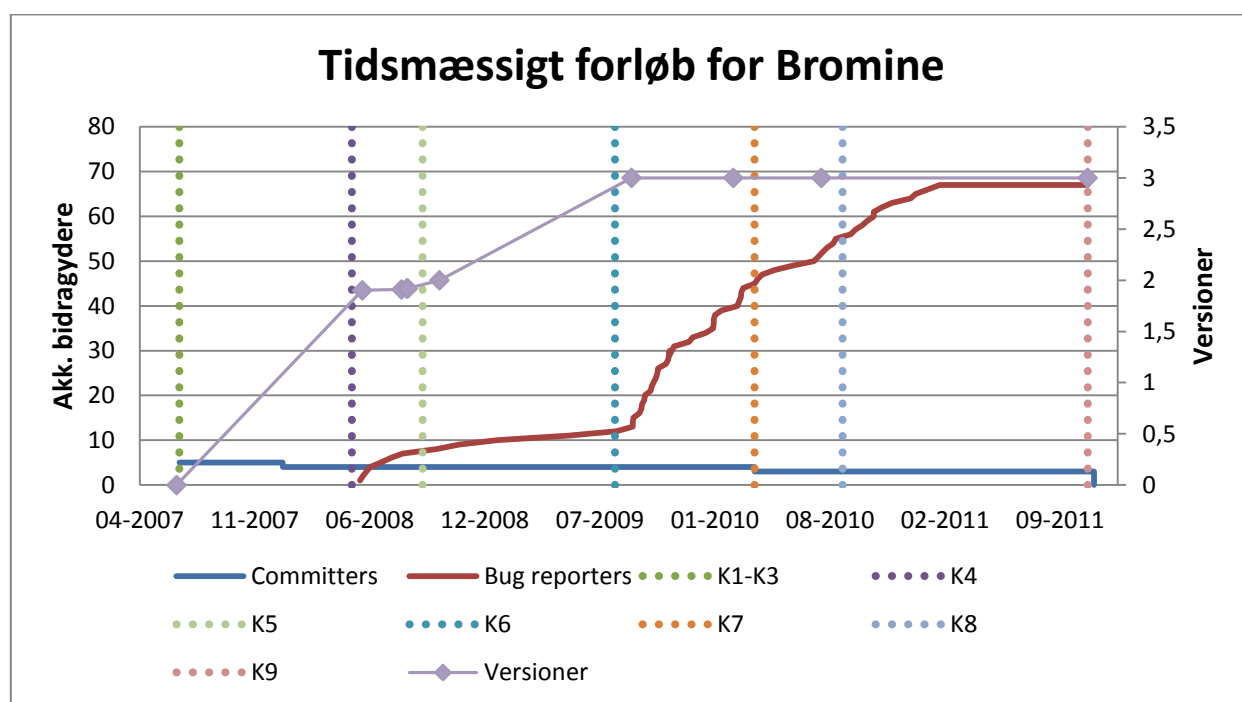
10.4.6 K9 - Projektet stoppes

Inden det blev besluttet at projektet skulle stoppe, var det allerede klart for udviklerne, at det ville blive sidste udgivelse, hvis de valgte at fortsætte projektet. I fællesskab blev det besluttet at stoppe projektet. Dette var primært baseret på, at ingen af udviklerne brugte Bromine længere, samt at projektet ikke længere var læringsmæssigt interessant, da det reelt handlede om at rette fejl i softwaren. De muligheder, der blev diskuteret, var bl.a. at lave en ny version, som inkorporerede nogle grundlæggende anderledes ideer. Dette ville have betydet en komplet omskrivning af kodebasen. Det afgørende argument for ikke at fortsætte var, at selvom de omskrev hele kodebasen, ville ingen af udviklerne bruge softwaren, da de alle havde skiftet job, hvor Bromine ikke kunne anvendes. Da udviklerne stadig havde en forpligtigelse overfor com-

muntyet, blev det forsøgt at overdrage projektet men uden held. Dette valg fjernede projektets levegrundlag og har derfor været essentielt for projektet.

10.4.7 Kritiske valg sammenfattet

De valg⁷¹, der blev taget i dette projekt, var alle været underlagt en kultur som ikke kunne bære et OSS projekt efter det mistede den kommercielle support fra Eniro. Sidste udgivelse var kort efter udviklernes opsigelse, og resten af indsatsen blev herefter overført til support. Kulturen var desuden med til at modvirke projektets vækst i den forstand, at det ikke var muligt at bibeholde andre udviklere end de oprindelige fra Eniro. Projektet havde ellers rig mulighed for at opnå en bred brugerdeltagelse, idet projektet havde adgang til et stort community i forbindelse med optagelsen ved OpenQA. I Figur 8 ses det tidsmæssigt forløb for Bromine og i **Fejl! Henvisningskilde ikke fundet.** vises en oversigt over kritiske valg.



Figur 8 Oversigt over kritiske valg set i forhold til projektets vækst⁷². De prikkede linjer angiver, hvornår et kritisk valg er opstået eller realiseret og skal ikke læses i forbindelsen med y-akserne. De solide streger viser udviklingen over tid for henholdsvis committers og bug reporters. De skal læses i forhold til den venstre y-akse. Diamanter indikerer tidspunktet for en given versions udgivelse. Når der er flere diamanter ud fra samme version, højre y-akse, skyldes det beta versioner.

11 Varnish Cache

Varnish Cache var et værktøj til at øge hastigheden på webapplikationer. Projektet startede i primo 2006 som følge af en udfordring, Verdens Gang (VG)⁷³ stod overfor. Udfordringen udmøntede sig i et trafikpro-

⁷¹ Se Bilag 21 for en oversigt over Bromines kritiske valg.

⁷² Bemærk at denne graf ikke indeholder authors, da dataene ikke fandtes.

⁷³ VG, Verdens Gang er en norsk medievirksomhed.

blem i forbindelse med udvidelsen af deres online del. Tidligere benyttede virksomheden Squid som caching løsning, hvor de havde 12 maskiner stående, som kørte denne applikation. For at finde en ny løsning gik VG og Linpro⁷⁴ sammen om at lave en løsning, som kunne opfylde VGs behov. Fra starten af projektet blev det fastlagt, at softwaren, der ville komme ud af projektet, skulle være open source. Denne beslutning skulle ses i lyset af, at VG ikke så sig selv i software branchen, og derved kunne Linpro få en forretningsmulighed. Denne mulighed resulterede i Varnish Software, som solgte komplementer ydelser til Varnish Cache. I forbindelse med udviklingen af softwaren blev PHK hyret som arkitekt og udvikler. På undersøgelsestidspunktet var det stadig primært PHK, som står for udviklingen af Varnish, som via sin Varnish Moral Licens (VML) bliver betalt for en del af hans udviklings tid. Omkring projektet var kommercielle interessenter bl.a. Fastly og Varnish Software. Disse virksomheder stod for en mindre del af udviklingen, og yderligere drev Varnish Software Varnish Caches hjemmeside.

Denne analyse har taget udgangspunkt i open source delen, dvs., Varnish Software er afgrænset til kun at blive optalt, hvor det har haft direkte relation til open source delen.

11.1 Organisation

Projektet bestod af seks personer. Der er taget udgangspunkt i, hvem der har haft commit adgang til projektet. I nedenstående graf er der vist strømmen af udviklere fra projektets start og frem til 23. april 2012, hvor dataene er hentet. En udvikler er anset som aktivt medlem i perioden fra første commit og frem til sidste commit. Som udgangspunkt har det været nødvendigt, at personen har været aktiv inden for de sidste tre måneder.

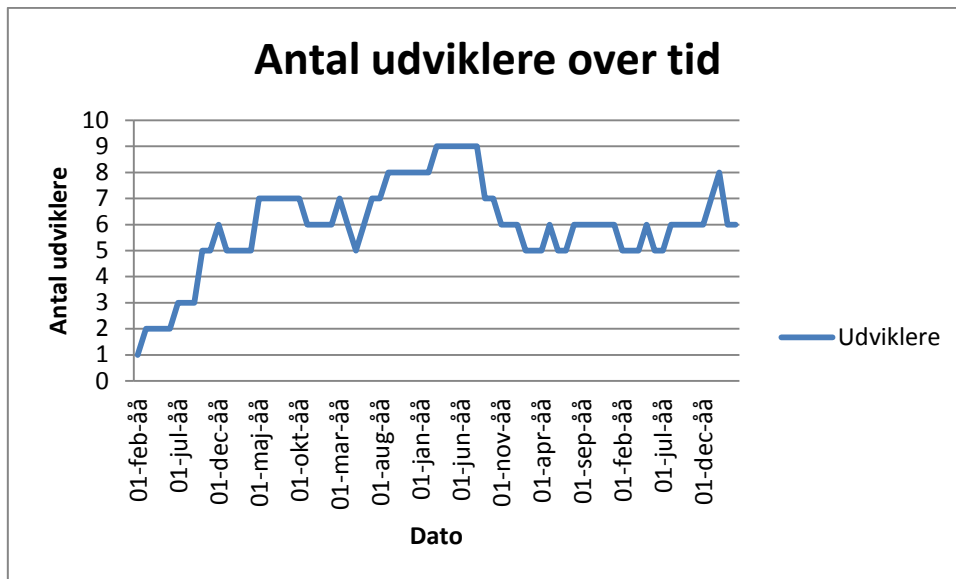
Projektet var kendetegnet ved, at det primært var PHK som skrev koden. Han stod for i alt 72 % af samtlige bidrag, der blev committet⁷⁵.

De seks personer, i Varnish projektet, var fordelt ved, at fire personer var fra Varnish software, en var betalt / frivillig⁷⁶ udvikler (PHK), og en var frivillig (APJ), som bl.a. havde påtaget sig opgaven at skrive dokumentation.

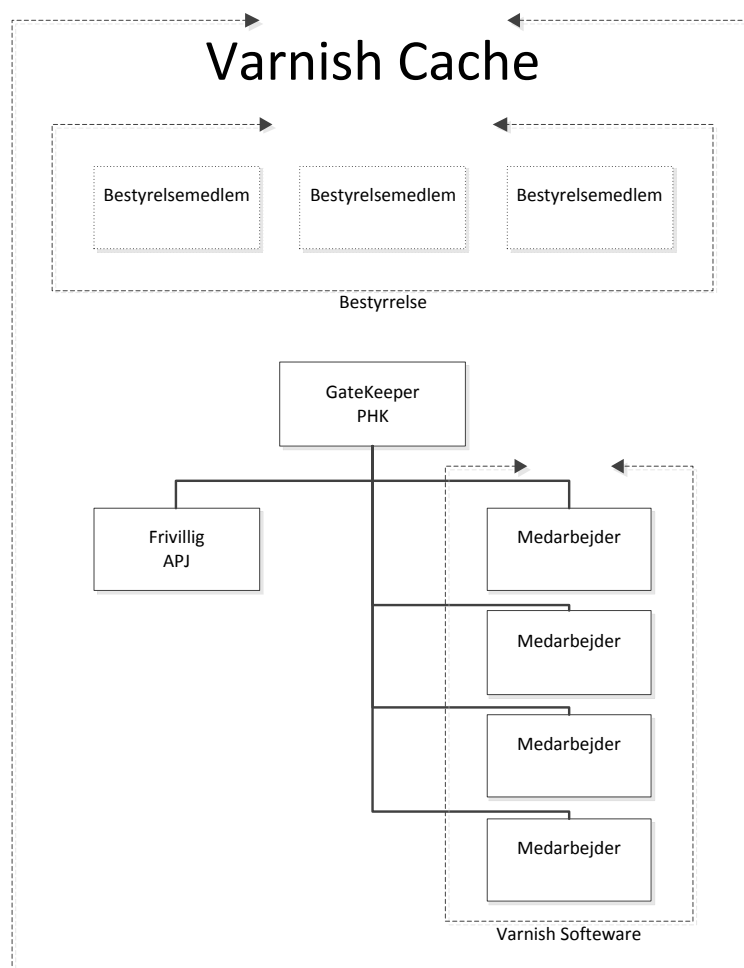
⁷⁴ Linpro blev i 2008 opkøbt af Redpil og hedder i dag Redpil Linpro.

⁷⁵ Se Bilag 25 for oversigt over antal bidrag

⁷⁶ Frivillig i den forstand at der blev brugt flere timer på projektet end der blev faktureret.



Figur 9 Graf over antal udviklere over tid (Defineret fra første og sidste commit).

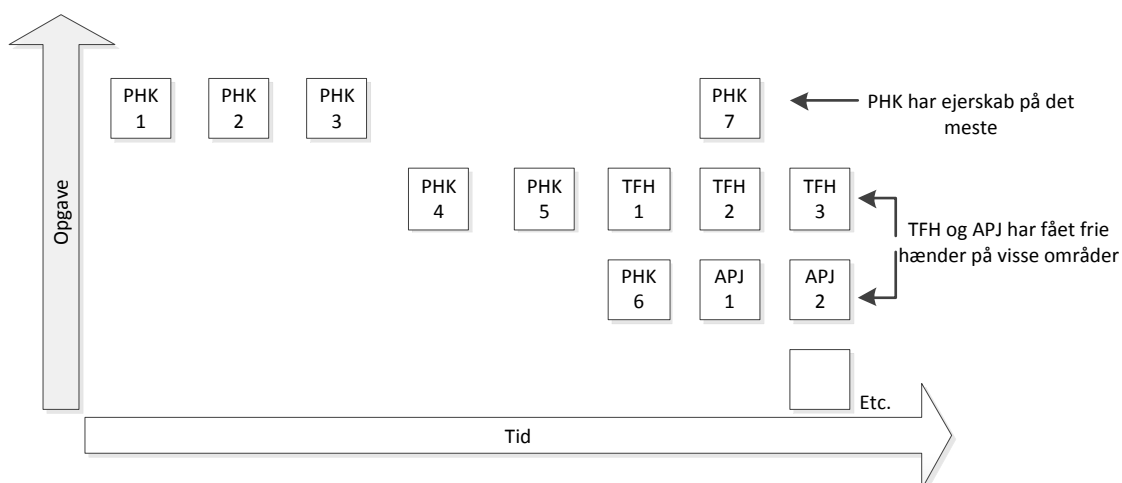


Figur 10 Varnish organisationsdiagram

Som organisationsdiagrammet har skitseret, var det PHK, som var gatekeeper for projektet. Når personer var interesserede i at blive en del af projektet og havde vist deres værd, fx via patches, var det også ham, som gav dem adgang til projektet. Varnish Software var modstykket til open source delen. Deres interesse i projektet var at kunne drive kommerciel virksomhed på baggrund af Varnish Cache. Det var ligeledes Varnish Software, som stillede servere til rådighed for projektet, hvilket gav dem en direkte adgang til projektet. Yderligere bidrog de økonomisk til PHK for hans udviklingstid på projektet. Det var desuden planlagt, at der skulle oprettes en bestyrelse bestående af tre personer. Bestyrelsen skulle blandt andet fungere som kontravægt til de kommercielle interesser, samt varetage projektets interesser, som et OSS projekt.

11.1.1 Arbejdsdeling

Oprindeligt var det VG i samarbejde med Linpro, som opsatte rammerne for udviklingen af softwaren, og det var PHK, som skulle implementere løsningen. PHK fik frie hænder til at implementere løsningen, som han ville. Senere i projektet, da VG havde fået deres løsning, var resten op til open source communityet. Efterfølgende var det stadig PHK som mere eller mindre enerådigt valgte, hvilke features der skulle implementeres, og hvilken løsning, der skulle anvendes. Situationen var speciel i den forstand, at PHK ikke selv var bruger af softwaren, hvilket betød, at hans eneste input var fra communityet, samt nogle kommercielle interessenter. Denne type organisation ligger sig mere op ad katedral organisationen. Arbejdsdelingen er skitseret som følgende.



Figur 11 Visuel repræsentation af Varnish arbejdsdeling^{77 78}

Organisationen var styret nedefra og op, da det var input fra communityet, der udmøntedes i features. Projektet var til gengæld meget centraliseret, hvor PHK sad på størstedelen af kodebasen og var beslutningsta-

⁷⁷ TFH er Tollef Fog Hagen, som var en af medarbejderne i Varnish Software.

⁷⁸ Dette er en simplificering af arbejdsdelingen for illustrativt at vise, hvordan organisationen så ud i forhold til katedral- og basarorganisationen.

ger for projektet. Der var dog visse områder, hvor andre har fået frie hænder. Her kunne nævnes dokumentationen, som varetoges af APJ, og nogle af områderne inden for hjælperprogrammer varetoges af TFH. Det var også PHKs opgave at gennemse alt kode, der blev indgivet til projektet. Andre opgaver som fx udgivelser og distributioner udførtes af medarbejdere fra Varnish Software.

11.2 Kultur

11.2.1 Artefakter

11.2.1.1 Fysiske udtryk

FreeBSD licens

(Mål). Valget af FreeBSD licensen udtryktes i forhold til de mål projektet arbejdede mod. Varnish skulle være en forretningsmulighed. Denne mulighed afspejledes i Varnish Software.

Pluggable interface

(Mål). VMods var en måde at åbne op for projektet, så andre havde nemmere ved at bidrage. Alternativet var tidligere at benytte inline C kode, hvor VMods var en pænere måde at gøre det på. VMods blev introduceret i version 3⁷⁹.

Nemt at komme i gang med

(Mål) Et mål i Varnish var, at det skulle være nemt at komme i gang med. Dette afspejledes i den måde Varnish var distribueret på. Til de understøttede platforme var det muligt at anvende et pakkestyringsværktøj til installationen og fjernelsen af Varnish.

Kontakt til community

(Midler, måling, korrektion). Kontakt til communityet spillede en stor rolle for Varnish. Her var VUG⁸⁰ en vigtig brik, da dette var et mødested for brugerne af softwaren, hvor man havde mulighed for ansigt til ansigt at tale med hinanden. Communityet spillede en afgørende rolle for måling af softwaren, da PHK ikke selv var bruger af softwaren. Via feedback fra communityet var det muligt at vurdere, om et fix eller feature levede op til forventningerne hos brugerne. I forhold til korrektion har communityet også indflydelse, da de gennem dialog med projektmedlemmerne kunne influere på deres valg.

⁷⁹ Se Bilag 7

⁸⁰ Varnish user group.

Gatekeeper

(Gruppegrænser). Projektet bar øjensynlig præg af, at det var PHK, som havde ansvaret for projektet. Al kode blev gennemlæst af PHK, og hvis man havde et ønske om at blive en del af Varnish holdet, skulle man igennem PHK først. Det var ham, som fungerede som en gatekeeper til projektet.

Lille organisation

(Gruppegrænse) Organisationen var relativt lille, da den bestod af seks personer. Da man talte med medlemmerne kom det også frem, at pladserne også var begrænset. Der var ikke plads til 20 udviklere, da det ikke var muligt at have flere, der kodede på de samme dele af systemet, grundet dets kompleksitet.

Commit fordeling

(Magt og status). Da man så på git historikken, var det klart, at det var PHK, som sad på størstedelen af kodbasen. Det var PHK, som havde magten til at bestemme fx hvilke patches, der blev committet.

Funktionsorganisation

(Ledelse, standarder). Projektet var funktionsopdelt, hvor det bl.a. var Varnish Software, som stod for at lave nye releases og pakke softwaren til distribution. I forhold til ledelse var det PHK, som stod i spidsen for projektet.

11.2.1.2 Sprog

Sweitzerkniv

(Mission). Projektet gik fra at være en caching løsning til at være en http sweitzerkniv.

Bug wash

(Midler). Bug wash var et møde, der blev afholdt hver mandag, hvor formålet var at diskutere de nye bugs, der var blevet tilføjet til projektets bug tracker.

Engelsk uformelt og teknisk talesprog

(Fælles sprog, ledelse). Sproget var primært engelsk og uformelt og bar præg af, at det handlede om en teknisk løsning, hvor fagsprog ofte indgik.

Enevælde

(Magt og status). PHK beskrev sin position i Varnish projektet som et enevælde, hvor han havde ansvaret som arkitekt på projektet.

11.2.1.3 Teknologi

Begrænset antal moduler

(Mål). Et mål med projektet var, at det kun var hyppigt brugte moduler, der blev en del af produktet.

Nemt at komme i gang med

(Mål). Målet "at det skal være nemt at komme i gang med" blev yderligere forstærket i og med, at hvis man ønskede at bruge Varnish, skulle man kun køre én fil for at starte programmet. Så var man i gang, men en god forståelse af softwaren var dog påkrævet for at få en optimal brug.

E-kommunikation

(Midler) Nogle af de midler, der blev brugt til at kommunikere, var internetbaserede løsninger som IRC, mailingliste og forum. De spillede en væsentlig rolle for projektet både internt men også for at kunne kommunikere med communityet.

Bug tracker

(Måling). En måling, som blev brugt, var i forbindelse med antal bugs. Medlemmerne i projektet bestræbte sig på at begrænse af antal åbne tickets i systemet.

Begrænset antal kontributioner

(Gruppegrænser). Medlemmer af projektet berettede, at projektet stillede høje krav til de udviklere, som potentielt kunne få adgang til systemet, da koden var svær at gå til. Dette kom til udtryk i form af de begrænsede antal kontributioner, som gennem årene blev givet til projektet.

Commit adgang

(Magt og status). Commit adgang i projektet gav en person status af at være en del af kernegruppen. Det var meget få personer, der fik commit adgang.

Performance

(Standarder). En af de standarder, der var at finde i projekt omhandlede performance målet, som nye releases og fixes hele tiden ville blive vurderet på.

Kodehygiejne

(Standarder). Kodehygiejne handlede om, at der ikke lå stykker kode, som ingen vidste, hvad gjorde, samt en stram tilgang til test hvor der var omkring 85 % testdækning.

11.2.1.4 Tradition

Varnish user group

(Måling). VUG var en halvårlig tradition, som blev introduceret i september 2009⁸¹.

11.2.1.5 Historier

Bestyrelse

(Korrektion, ledelse). Bestyrelsen var aldrig blevet til noget, selvom det var bestemt, hvem der skulle bestrebe den.

11.2.2 Værdier

"Vi skal være de bedste"

(Mål). Projektet var gennemsyret af, at de ville være de bedste. Der stilledes store krav for at kunne blive en del af projektet, og alt kode blev gennemset af PHK, før det blev committet.

Få der kan løfte opgaven

(Gruppegrænser). Så man på organisationen, var den relativt lille. Dette stammede delvist fra, at der i projektet herskede en overbevisning om, at det var et fåtal, som kunne overtage dele eller hele projektet og køre det videre på et fornuftigt plan.

"Vi har bevist vores værd"

(Magt og status). Der var en tiltro til, at personer, som var en del af projektet, havde bevist deres værd. PHK blev indhentet til projektet på baggrund af hans tidligere arbejde med FreeBSD projektet. APJ havde, før han fik commit adgang, lavet nogle patches til projektet og mente, at hans tidligere arbejde havde haft en indflydelse på hans hurtige optagelse i projektet.

11.2.3 Grundlæggende antagelser

"Vi er udsatte"

Denne grundlæggende antagelse udsprang fra, hvorledes projektet styredes. Projektet blev primært styret og drevet af PHK. Busfaktoren⁸² i projektet var en, hvilket ville sige, at projektet var meget udsat, hvis PHK ikke længere kunne være en del af projektet. PHKs arbejde blev hovedsageligt drevet af salg af hans Varnish Moral License, som blev brugt til at finansiere hans tid på projektet. Set på de andre medlemmer og personer som tidligere havde været involveret, bar det præg af, at folk oftest kun kort var en del af projektet. Dvs., når deres problem var løst, var de væk igen. Varnish software havde en kommerciel interesse i projek-

⁸¹ Se Bilag 8

⁸² Busfaktoren er en måling af hvor mange nøglemedarbejder der må fjernes fra projektet før at projektet går i stilstand.

tet, hvilket betød deres engagement udsprang fra en indtjeningsmulighed. APJ blev en del af projektet i forbindelse med hans arbejde, og efterfølgende blev han hængende, fordi han syntes, det var interessant at være med, hvis projektet gik i stilstand, ville hans engagement formentlig være udsat.

Udviklerne skal være eksperter

At udviklerne skulle være eksperter, afspejledes i organisationens størrelse og måde, hvorpå kode blev committet. Selvom der var flere niveauer, man kunne være med på, så skulle man være ekspert, hvis man skulle være en af udviklerne pga. softwarens kompleksitet.

"Vi skal være de bedste"

Selvom denne sætning var nævnt som en værdi, var det også en grundlæggende antagelse i projektet. Det gennemsyrede hele projektet, at udviklerne skulle være de bedste til det, de gjorde. Dette hang sammen med personernes motivation, som måske ellers ikke var interesserede i at være en del af projekt, hvis dette ikke var en udtalt værdi. Dette kunne også ses på de mål, der omtaltes, hvor performancemålet direkte hang sammen med, at de ville være de bedste, og det skulle være muligt at presse mest muligt performance ud af et givent system. Yderligere havde det givet Varnish Software mulighed for at kunne drive forretning på denne softwareløsning, og (samt flere andre virksomheder) var med til at købe Varnish Moral License, som havde finansieret projektet.

"Vi gør det vi er bedst til"

Antagelsen om, at "de gør det de er bedst til" kunne udtrækkes i forhold til den funktionsopdeling, der var. Dette betød ikke, at projektmedlemmerne afholdtes fra visse opgaver, men at de mest egnede personer løftede de opgaver, som de var bedst til, eller havde tid til, så andre kunne koncentreres sig om de opgaver, de var bedst til.

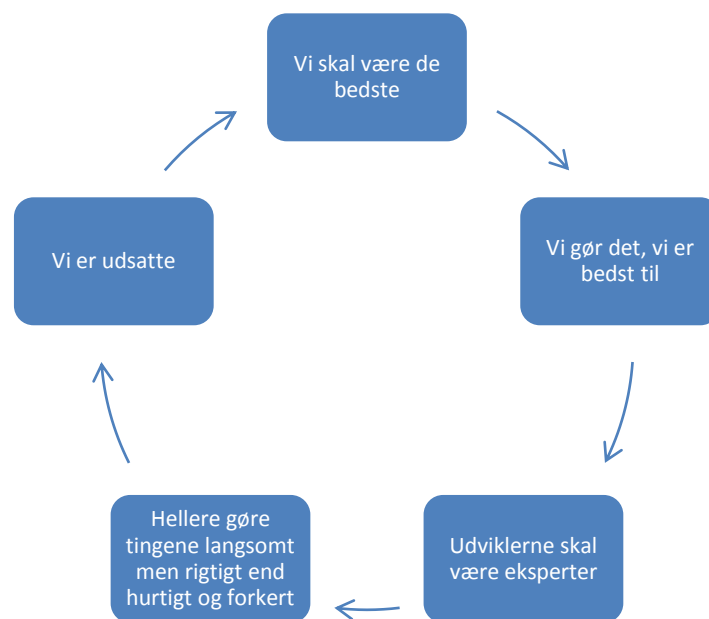
Hellere gøre tingene langsomt, men rigtigt end hurtigt og forkert

I projektet var der en antagelse om, at tingene hellere måtte tage den tid, der var nødvendig og dermed med større sandsynlighed løse opgaven rigtigt første gang. Dette kunne også relateres til det høje optagelseskrav, og at PHK var gatekeeper for projektet. Hvis ikke denne antagelse var gældende, ville projektet sandsynligvis bestå af flere udviklere uden de strikse optagelseskrav. Hvilket kunne betyde hurtigere løsninger, men som måske ikke kunne leve op til performanceidealet, som projektet styredes af.

11.2.4 Det kulturelle paradigme

Det kulturelle paradigme for Varnish er skitseret i Figur 12. I figuren er der vist et cirkulært afhængighedsdiagram, som begynder med antagelsen om, at de skulle være de bedste. Denne antagelse hang sammen med, at de forsøgte at gøre det, de var bedst til, da dette ville give en så vidt mulig optimal arbejdsdeling.

At de ønskede at gøre det, de var bedst til, henleddes til det faglige niveau lagt af PHK i form af hans arbejde, som kom til udtryk i antagelsen om, at udviklerne skulle være eksperter for at kunne blive en del af projektet. Med den ekspertise og de begrænsede ressourcer, der var tilgængelige for projektet henleddes til antagelsen om, at de hellere ville gøre tingene langsomt men rigtigt frem for hurtigt og forkert. Kombinationen af disse antagelser resulterede i en kultur, hvor det var antaget, at projektet var særlig sårbart og udsat. Resultatet var, at de stadig ønskede at være de bedste, men det blev en selvforstærkende effekt, hvor projektet blev endnu mere udsat, da enkelte nøglepersoners forsvinden kunne have fatale konsekvenser for projektet.



Figur 12 Varnish kulturelle paradigme

11.3 CBPP processen

Projektet var beskyttet i form af FreeBSD licensen. Denne licens gav alle mulighed for at benytte softwaren, som de havde lyst til. Dette indebar også, at man ikke skulle gøre produkter, som var direkte nedarvede eller modificerede udgaver af den oprindelige software, tilgængeligt for andre med samme rettigheder. Der fandtes dog nogle sociale normer⁸³, som gjorde, at nogle alligevel valgte at give tilbage til projektet fx Fastly, som havde deres egen branch af Varnish. Denne licens gjorde, at alle kunne have en interesse i at gøre softwaren bedre og havde mulighed for det i forhold til de webbaseret teknologier, som projektet benyttede.

⁸³ Disse normer var ikke genstand for analyse, da det blev afgrænset i forbindelse med denne afhandling fra direkte at inddrage den omkringliggende brugergruppe. Det kunne dog konstateres, at disse normer fandtes både igennem interview med personer i kernegruppen men også ved at observere, hvem der var forfattere til de commits, der blev en del af produktet.

de sig af, hvilket betød, at projektet havde alle forudsætningerne for at kunne drage nytte af ressourceallokeringsfordelen.

Det potentielle, der lægger i ressource allokeringensfordelen, kan blive realiseret, når projektet udnytter de muligheder, der ligger i informationsbehandlingsfordelen, som munder ud i en bred brugerdeltagelse og filtrering af kontributionerne til projektet. I dette projekt var der en lille brugerdeltagelse set på kernegruppen, hvor de involverede personer havde hver deres rolle. Der var flere faktorer, der påvirkede muligheden for en bredere brugerdeltagelse. Dels var der en kultur, som begrænsede muligheden for flere medlemmer i og med, at der blev stillet store krav til de involverede personers fagkundskaber, når det handlede om udviklere. Selve arkitekturen af softwaren spillede også en væsentlig rolle, da Varnish kildekoden var svær at gå til. Det hang sammen med kulturen, og personer, som ville involvere sig skulle besidde en hvis mængde ekspertise for at kunne løfte opgaven som udvikler. Der var dog siden version 3 blevet åbnet op for arkitekturen ved at introducere VMods, hvilket resulterede i 18 VMods⁸⁴. Selvom projektet kun havde et begrænset antal medlemmer, havde de kontakt til communityet, som var med til at give input til projektet. For at filtrere inputtene og kontributionerne blev de diskuteret internt i projektet, men i sidste instans var det hovedsageligt PHKs opgave at implementere disse løsninger eller gennemse koden, før den blev commitet.

Både ressourceallokerings- og informationsbehandlingsfordelen kræver, at man overkommer motivationsudfordringen. Projektet blev primært drevet af ydre motivationsfaktorer, hvor de involveret havde en kommerciel interesse i projektet. Varnish Software forsøgte at drive virksomhed på baggrund af det udviklede software. De to frivillige var begge blevet en del af projektet i forbindelse med deres arbejde. De frivillige motiveredes ikke kun gennem ydre motivationsfaktorer men også af glædesbaseret motivation som fx muligheden for at lære noget nyt eller blot, at de syntes, det var en interessant kontekst, hvori softwaren benyttedes. Ud fra dette kunne det konstateres, at projektet ikke udnyttede alle de motivationstyper, der var mulighed for at benytte sig af. Skalerbar motivation var ved at blive en realitet, da projektet var blevet mere åbent med introduktionen af VMods. Der blev ikke identificeret nogen kontekstbestemte motivationsfaktorer, hvilket kunne være en af forklaringerne på, at mange af dem, som kortvarigt valgte at involvere sig i projektet, forsvandt, når deres problem var løst.

Når ressourceallokerings- og informationsbehandlingsfordelene er håndteret, opstår der et behov for koordineret handling, som er en del af organisationsudfordringen. Som beskrevet i afsnittet om arbejdsdeling blev projektet koordineret centralt, hvor det var PHK, som i sidste instans var beslutningstager med input

⁸⁴ Se Bilag 9

fra communityet og de andre projektmedlemmer. Da projektet var funktionsopdelt, havde medlemmerne en primær rolle, som hovedsageligt blev varetaget af den enkelte. Selvom projektet internt var funktionsopdelt, betød det ikke, at medlemmer ikke kunne arbejde på andre dele af projektet. Denne funktionsopdeling var ikke et bevidst valg men resultatet af kulturen i projektet. Der var som beskrevet i afsnittet om organisationen flere udviklere tilknyttet projektet både i kernegruppen og den aktive del af communityet, som arbejdede på forskellige løsninger til projektet, hvilket indikerede en flygtighed i projektet, som havde potentiale for at kunne drage bedre nytte af både ressourceallokerings- og informationsbehandlingsfordelen.

Efter opgaverne i projektet var blevet koordineret, opstod behovet for integration. Som allerede nævnt var det PHK, som valgte, hvilke løsninger, der skulle implementeres med feedback både fra de andre medlemmer og communityet. Med hensyn til udgivelser var det primært Varnish Software, som havde ansvaret, når det kom til at definere versionsnummer og sørge for, at de nye versioner var nemt tilgængelige via hjemmesiden.

11.4 Kritiske valg

11.4.1 K1+ K7 - Licensvalg

At projektet blev udgivet under en permissive licens betød en større mulighed for samarbejde med kommercielle interessenter. Ved at anvende denne licens havde virksomheder mulighed for at benytte softwaren som en konkurrencemæssig fordel, hvorimod hvis det havde været en GNU GPL, så ville virksomhederne være tvunget til at frigive deres kode. Selvom virksomhederne havde mulighed for at have deres egen private branch, ville der være en fordel i at samarbejde med OSS delen, da en meget modificeret branch ville gøre det kompliceret at drage nytte af opdateringer fra OSS delen. K1 er derfor betegnet som havende indflydelse på en bredere brugerdeltagelse.

Konflikten imellem APJ og Varnish Software vedrørende valg af licens til Varnish bogen betød, at et samarbejde mellem parterne ikke var muligt. Dette ville betyde en ulempe for projektet, da det ville betyde redundans og derved øge vedligeholdelsesomkostningerne. K7 er derfor betegnet som havende negativ indflydelse på CBPP processen. Denne negative indflydelse ville betyde en forringelse af den koordinerede handling.

11.4.2 K2 - Kultur - Vi skal være de bedste

På grund af kulturen bestod communityet primært af aktive brugere med en lav deltagelse af perifere udviklere. Indtil videre havde det ikke været et problem for projektet, da projektet blev drevet af ydre motivationsfaktorer. Udfordringen ville opstå, når det økonomiske incitament fjernes. I så fald ville PHKs enga-

gement stoppe, og projektet ville stå uden kerneudviklere. Hvis projektet ikke formåede at ændre kulturen i tiden eller finde den helt rigtige til at overtage projektet, ville ansvaret lande på de kommercielle interesser. Kulturen ville være en hæmsko i forhold til at finde nye til projektet, da projektets adgangskrav var sat højt. Derfor kunne valget endnu ikke betegnes for kritisk, men det havde et potentiale, som ville have en afgørende betydning for projektet. Selv hvis de fandt den rigtige til at overtage projektet, ville det blot være et spørgsmål om tid, før projektet ville risikere at stå i samme situation igen. Set i forhold til CBPP processen var kulturen med til at reducere den brede brugere deltagelse i forhold til perifere udviklere og kerneudviklere.

11.4.3 K3 - Distributioner

En af målsætningerne i projektet var, at det skulle være nemt at installere. Derfor havde distributionerne spillet en vigtig rolle i forhold til, hvordan brugerne kunne installere softwaren. Ved at vælge distributionskanaler, som de pakkehåndteringssystemer der var integreret i Linux miljøerne, havde det været muligt at udgive softwaren, så brugeren blot kunne køre nogle få kommandoer og dermed få Varnish til at fungere. Ved at gøre det nemt for brugeren at få adgang til softwaren betød det dels, at en mængde supportspørgsmål ville blive trivielle at svare på, men også at det ville øge udbredelsesmulighederne, da kravene til brugeren derved var mindre. I forhold til CBPP processen var det et vigtigt valg, fordi det påvirkede, hvorledes projektet blev gjort tilgængeligt for brugerne.

Mange valgte at bidrage som aktive brugere ved at rapportere fejl, og det var en indikation på, at der var mange, som benyttede softwaren, og gerne ville være med til at gøre den bedre. Hvis softwaren havde været svært tilgængelig via en besværlig installationsprocessen, kunne det have påvirket projektet negativt i forhold til væksten af communityet.

11.4.4 K4 - Kode hygiejne

En af de ting PHK havde taget med sig fra hans tid i FreeBSD projektet, var en stram teststrategi, samt hvordan koden skulle dokumenteres. PHK kaldte det selv for kode hygiejne, hvor det var vigtigt, at der ikke fandtes kode, som ingen vidste, hvad gjorde. Denne tilgang betød, at i de tilfælde, hvor der blev foretaget ændringer i koden, var en større tiltro til at ændringer ikke havde påvirket softwaren negativt. Dette fik en væsentlig betydning for projektet, da PHK ikke selv var bruger af softwaren, og dermed ikke selv kunne teste ændringerne på et produktionsmiljø.

I forhold til CBPP processen var dette ikke noget, som blev behandlet, i stedet handlede det primært om peer-review⁸⁵, når der skulle diskuteres metoder til kvalitetssikring. På grund af projektets art, hvor antallet

⁸⁵ Peer-review er processen, hvor communityet tager del i at gennemse koden, før den bliver committet.

af udviklere var begrænset, spillede kodehygiejne en væsentlig rolle, da det hovedsageligt kun var PHK, som gennemså koden. Derfor var kodehygiejne ikke kun et middel til kvalitetssikring men også en metode, som kunne være et supplement til en ellers manglende evne til at udføre peer-reviews.

11.4.5 K5 - VUG

VUG var et alternativ til de e-kommunikationsmidler, som man normalt kunne finde i et OSS projekt. Det var vigtigt at afholde møderne, da det var en måde at sætte brugerne sammen, så de kunne udveksle viden. En viden, som ville have været svære at dele via online kommunikation, da en personlig kontakt kunne være med til at skabe en mere uformel dialog blandt brugerne. E-kommunikation blev ofte brugt til problemløsning⁸⁶, hvorimod et brugermøde ville give anledning til diskussioner omkring brugen af softwaren og dermed også succes historier.

Dette valg er betegnet som værende af væsentlig karakter for projektet. Ved at benytte brugergrupper giver det mulighed for en anden type bred brugerdeltagelse, end der blev refereret til i CBPP processen, som betegnede e-kommunikationsmidlerne som kritiske⁸⁷.

11.4.6 K6 - VMODs

VMODs var stadig nyt på undersøgelsestidspunkt, så derfor var det for tidligt at kunne konkludere, hvilken betydning det ville få for projektet. Der fandtes dog nogle indikatorer på, at det potentielt kunne få en væsentlig betydning for projektet. Før VMODs var der reelt ingen måde at bidrage til projektet, uden man satte sig ind i hele kodebasen. Før blev alt håndteret i VCL, hvor man kunne bruge inline C kode, hvis der ikke var noget, der opfyldte ens behov. Med introduktion af VMODs blev arkitekturen åbnet op, hvor man kunne udvikle sine egne moduler og dele dem med andre via Varnishs hjemmeside.

Set i forhold til CBPP processen havde den en afgørende betydning i forhold til, at softwaren skulle være modulær og granulær for at kunne imødekomme en bred brugerdeltagelse, som gjorde det muligt at drage nytte af skalerbar motivation, hvor selv lidt motivation kunne være nok til, at man bidrogede til et projekt.

På undersøgelsestidspunktet er det betegnet som et vigtigt valg for projektet med et potentiale for at kunne blive essentielt for projektet. Det kunne blive essentielt, da det kunne være med til, at flere personer ville sætte sig ind i kodebasen, og derved åbne op for muligheden for at nye perifere og kerneudviklere kunne blive en del af projektet.

⁸⁶ Uddrag af forum indlægsoverskrifter se Bilag 19

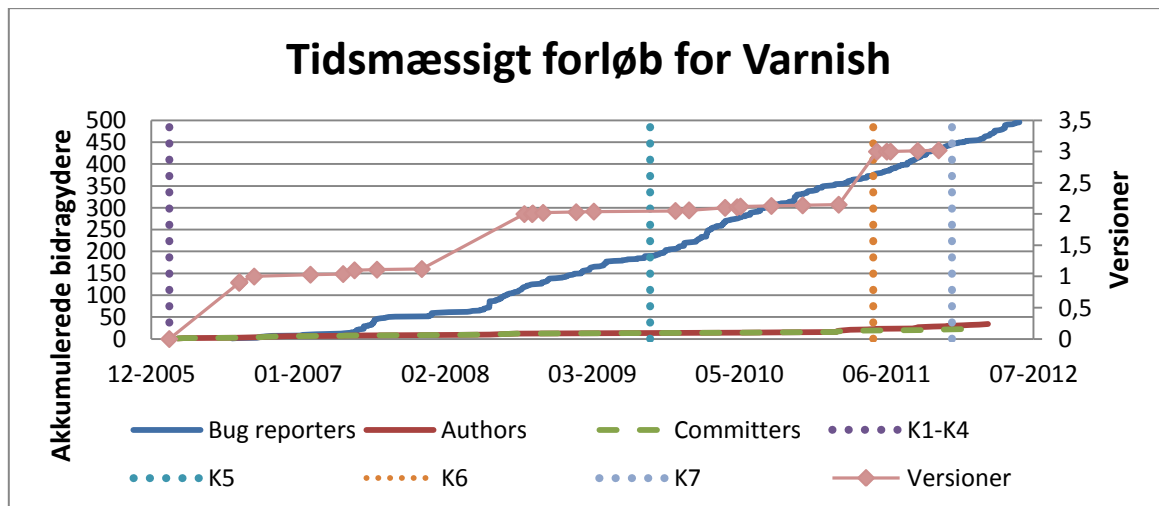
⁸⁷ (Andreev, Feller, Finnegan, & Moretz, 2010, s. 7)

11.4.7 K8 - Bestyrelse

Valget om, at der skulle indsættes en bestyrelse, ville have en potentiel kritisk værdi på fire. Bestyrelsens opgave skulle blandt andet være modstykket til de kommercielle interesser, og den ville spille en central rolle, når de overordnede rammer skulle sættes for projektet. På grund af PHKs involvering, hvor han havde modtaget betaling, og derved havde haft mulighed for at arbejde nær fuldtid på projektet, havde behovet for en bestyrelse været mindre. Dette reflekteres desuden i den centraliserede ledelse af projektet, hvor han havde varetaget mange af de opgaver, en bestyrelse normalt ville have varetaget. Bestyrelsen havde potentielt mulighed for at være essentiel for projektet.

11.4.8 Kritiske valg sammenfattet

Som det fremgår af Figur 13 er der fundet en indikation på, at de kritiske valg⁸⁸ har været med til, at projektet har kunnet fastholde en konstant vækst i projektets løbetid. Det interessante ved dette projekt har været forskellen på mængden af authors og bug reporters. Størstedelen valgte at deltage som aktive brugere modsat at et fåtal, der valgte at engagere sig som perifere udviklere. En forklaring på dette kunne findes i en kombination af kulturen og kompleksiteten af koden. Kodens kompleksitet stillede høje krav til nye udviklere. Dette afgrænsede naturligt en mængde udviklere fra at deltage. Dette blev reflekteret i kulturen, som var med til at holde kravene oppe. Hvis PHK stoppede, og der ikke var nogen, som var villig til at tage over, var der to muligheder. Enten stoppede projektet eller også skulle adgangskravene reduceres, og personer, som besad en mindre mængde ekspertise, ville få adgang til projektet.



Figur 13 Oversigt over kritiske valg set i forhold til projektets vækst. De prikkede linjer angiver, hvornår et kritisk valg er opstået eller realiseret og skal ikke læses i forbindelsen med y-akserne. De solide streger viser udviklingen over tid for henholdsvis committers, authors og bug reporters. De skal læses i forhold til den venstre y-akse. Diamanter indikerer tidspunktet for en given versions udgivelse. Hvis der er flere diamanter ud fra samme version, højre y-akse, skyldes det beta versioner.

⁸⁸ Se Bilag 22 for en oversigt over Varnishs kritiske valg.

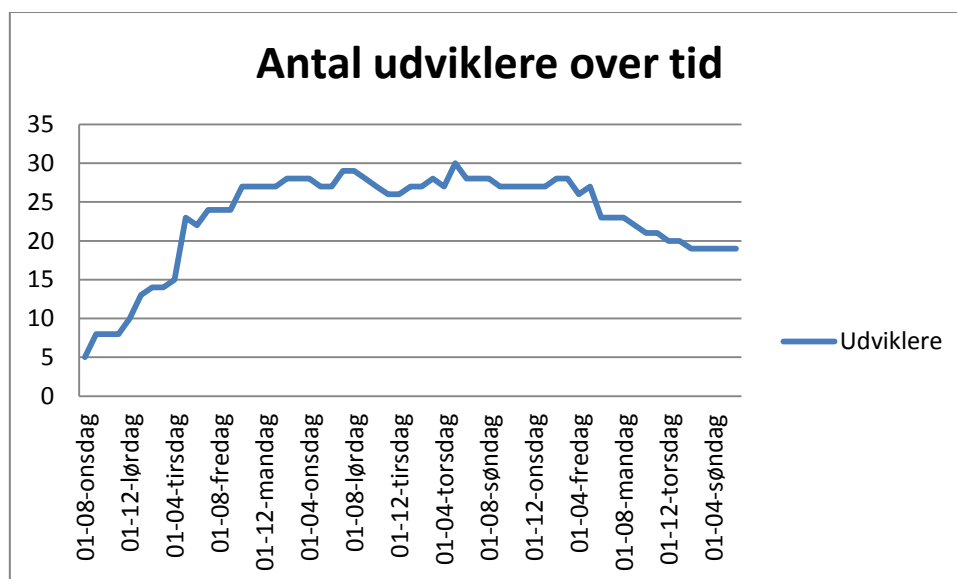
12 Exherbo

Exherbo var en Linux distribution, som blev startet i 2007 af Bryan Østergaard (BØ). Projektet startede som et nyt hobbyprojekt efter, han valgte at stoppe sit arbejde med Gentoo projektet. Gentoo var også en Linux distribution. Projektet blev betegnet som en legeplads for ideer, BØ sad inde med, mens han forsøgte at finde ud af, hvad han nu skulle bruge sin nyfundne tid på. Efterhånden som Exherbo udviklede sig, fik han tilkendegivelser fra sine omgivelser om, at det var en interessant distribution, han havde gang i. Efterhånden som tiden gik, måtte han indse, at han egentlig havde gang i en reel Linux distribution, selvom det ikke var hensigten at starte en ny distribution.

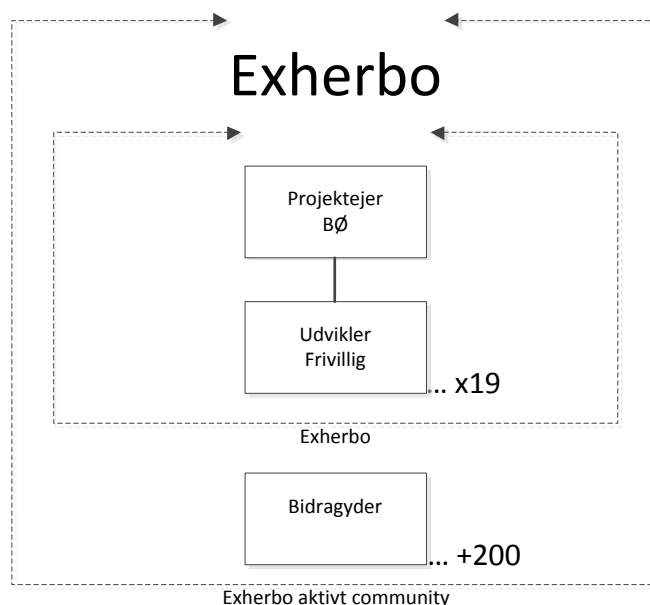
I skrevne stund er projektet vokset til at have 19 kerneudviklere og have modtaget kontributioner fra mere en 200 forskellige personer.

12.1 Organisation

Projektet var organiseret som et fladt hierarki bestående af en projektejer, en kernegruppe bestående af 19 personer + BØ, samt et aktivt community, hvor den perifere udviklergruppe talte over 200 forskellige individer i løbet af projektets levetid. På grund af communityets art var det ikke muligt at give et definitivt svar på det aktive communitys størrelse. Størrelsen betegnedes som flydende i det omfang, at der kontinuerligt kom nye personer til projektet samt at nogle personer ville falde fra, hvorfor det blev skitseret som antal kontributioner i hele projektets levetid. Det har været særligt relevant at inddrage denne dimension i dette projekt pga. den store andel af kontributioner modsat Varnish og Bromine, som primært blev drevet af nogle få individer. Figur 14 viser kernegruppens udvikling over tid. Organisationen er skitseret i Figur 15.



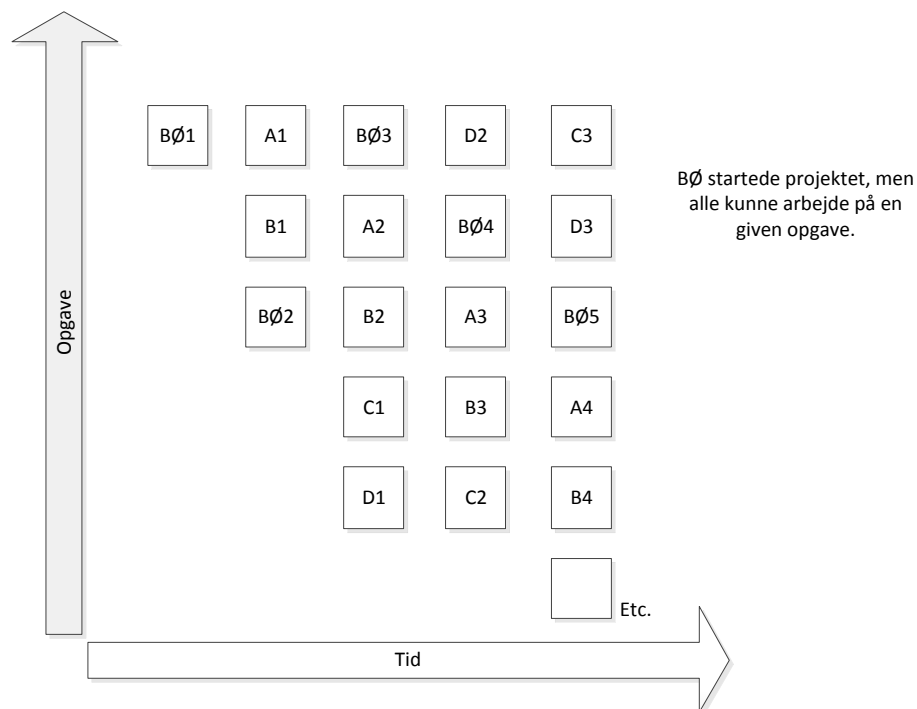
Figur 14 Graf over antal udviklere over tid. (Defineret ud fra første og sidste commit)



Figur 15 Exherbo organisationsdiagram

12.1.1 Arbejdsdeling

I begyndelsen af projektet var det alene BØ, som satte rammerne og definerede opgaverne for projektet. Senere begyndte andre at interessere sig for hans projekt, og de kom med på projektet. Projektmedlemmerne valgte selv, hvilke opgaver de ville arbejde på. Overordnet var det stadig BØ samt kernegruppen, som satte rammerne for projektet, men mange af beslutningerne blev taget demokratisk. Denne type organisation lægger sig meget tæt op af basarorganisationen med et fladt, internt magthierarki, hvor BØ sad som projektejer.



Figur 16 Visuel repræsentation af arbejdsdelingen i Exherbo⁸⁹

Organisationen var kendetegnet ved en decentraliseret organisering, hvor udviklerne arbejder på tværs af opgavernesamtidig med, at der ikke var nogen, som enerådigt havde magt over en given opgave.

12.2 Kultur

12.2.1 Artefakter

12.2.1.1 Fysiske udtryk

Hjemmeside

(Mission og mål). Exherbos hjemmeside indeholdte en række information, som blev brugt til at definere projektets mission og mål⁹⁰. Det blev nævnt bl.a. i forhold til mission, at de forbeholdte sig retten til at lave radikale ændringer i softwaren, hvis det var nødvendigt. Et mål med projektet var, at brugerne skulle deltage aktivt i projektet.

Afstemning af forventninger

(Midler). På projektets hjemmeside var der udformet en tekst, som klart og tydeligt beskrev, hvad projektet forventede af brugerne. Dette blev brugt som middel til bl.a. at realisere målet om, at brugerne skulle deltage aktivt, da det kunne frasortere nogle uønskede brugere på forhånd.

⁸⁹ Dette er en simplificering af arbejdsdelingen, for at illustrativt at visse hvordan organisationen ser ud i forhold til katedral og basar organisationen.

⁹⁰ Se Bilag 11 for specificering af disse informationer.

Kerneudviklere

(Magt og status). Kerneudviklerne var dem, som havde magt til at committe kode til deres CVS samt at tage beslutninger for projektet.

12.2.1.2 Sprog

Social engineering

(Midler, ledelse). Social engineering⁹¹ var, hvad BØ kaldte sin ledelsesform. Begrebet dækkede bl.a. over, hvordan man manipulerede folk til at deltage ved at stimulere deres motivation.

Social rangstige

(Magt og status). I forbindelse med BØs definition af social engineering fremgik det, at der fandtes en social rangstige inden for OSS, som kunne gøres op i aktivitet og commits.

Hård omgangstone

(Fælles sprog). Til tider kunne sproget i projektet betegnes som hård, da det var acceptabelt at kalde hinanden fx idioter. Påstanden om, at den anden var en idiot, skulle begrundes, da simpel tilsvining ikke var tilladt.

12.2.1.3 Teknologi

E-kommunikation

(Midler). Den primære kommunikation i projektet foregik via IRC. Projektet benyttede desuden en mailingliste.

Git

(Midler). Git var projektets VCS. Dette spillede en væsentlig rolle udover dens primære funktion som revisionskontrol system, da man kunne få sit navn på et commit, selvom man ikke havde commitadgang.

Kode review

(Måling). For at sikre kvaliteten af kontributioner anvendtes en patch-bot i projektes IRC kanal, der indeholdte en kø med kontributioner, som dernæst skulle gennemses for fejl, før de blev commitet.

Distribueret udviklingsmodel

(Ledelse). I projektet anvendtes en distribueret udviklingsmodel, som fremgik af afsnittet omkring arbejdsdeling.

⁹¹ For meget information omkring BØ's definition af social engineering se Bilag 13.

12.2.1.4 Tradition

Konflikthåndtering

Hvis der opstod konflikter, hvad enten det var tekniske eller sociale, var det kerneudviklernes opgave at finde en løsning. Hvis konflikterne ikke kunne løses, eskaleredes konflikten til BØ, som fandt en løsning.

Afvigelse fra værdier

(Gruppegrænser). Hvis man afveg for projektets værdier, kunne man risikere at blive udelukket fra projektet. Dette ville foregå igennem dialog, hvor det blev forklaret, hvorfor man ikke passede ind i projektet, og at der sikkert var et andet projekt, man ville passe bedre i.

Du skal bevise dit værd

(Gruppegrænser). For at blive optaget som kerneudvikler var der tradition for, at man skulle have bevist sit værd. Dette inkluderede, at man stod inde for projektets værdier og typisk skulle have været aktivt i en længere periode.

Alle må deltage i diskussionerne

(Magt og status). Der var tradition for at alle i communityet var velkommen til at deltage i projektets diskussioner.

12.2.1.5 Historier

Manifest

(Mission). Tidligt i projektet var der et manifest, som hårdt men ærligt meldte ud, at projektet ikke var interesseret i andres meninger eller deltagelse i projektet.

Kun et hobbyprojekt

(Mission). Til at starte med var projektet BØs hobbyprojekt, som det skulle erstatte den tid, han tidligere havde brugt på Gentoo projektet. Det var aldrig hans henseende, at starte et større projekt, men det skulle blot være et projekt, han selv kunne have det sjovt med.

(Midler). Deltagelse i konferencer

(Midler). Et afledt middel til at nå ud til nye omgangskræse var at deltage i konferencer. Dette gav projektet eksponering, som ellers ikke ville have været muligt.

12.2.2 Værdier

Kernegruppen må ikke blive for stor

(Mission). En af de værdier der blev lagt vægt på var, at projektet ikke måtte have for mange kerneudviklere, da dette kunne være med til at kompromittere projektets værdier.

Vi skal kunne stå inde for det tekniske, det skal være sjovt og lærerigt

(Mål). Projektet bestræbte sig på, at det skulle indeholde en vis mængde teknisk kvalitet. Det skulle yderligere været sjovt og lærerigt for de involverede.

Du skal være på IRC

(Midler). Det forventedes, at man brugte meget tid på IRC for at kunne følge med i diskussionerne.

Freeloaders må ikke blive en byrde

(Gruppegrænser...). Freeloaders i communityet var anset som værende acceptabelt, så længe de ikke var en supportbyrde for projekts medlemmer.

Du forventes at deltage aktivt

(Gruppegrænser). Det forventedes, at brugerne deltog aktivt i communityet.

Du forventes at læse dokumentationen

(Gruppegrænser) Det forventedes, at brugerne læste dokumentationen og satte sig ind i softwaren, før de tog kontakt til communityet.

Ingen bestemmer over andre

(Magt og status). I projekt blev alle anset som ligeværdige.

Tilkendegiv nytilkommendes kontributioner

(Magt og status, ledelse). For at højne nytilkommendes status i communityet blev der lagt vægt på at eksponere deres kontributioner og derved give dem en hurtig succesoplevelse.

Vi hjælper hinanden

I projektet var der lagt stor vægt på, at man skulle hjælpe hinanden især nytilkommende.

12.2.3 Grundlæggende antagelser

Vi er en homogen gruppe

Der var en grundlæggende antagelse om, at projektet var centreret omkring en homogen kernegruppe. Dette afspejledes i måden, personer kunne blive optaget i kernegruppen. Hvis personer afveg fra projektets værdier, ville personen blive afvist.

Vi gør det, fordi vi synes, det er sjovt

I projektet var det antaget, at det skulle være sjovt at arbejde på projektet. Dette var både en udtrykt værdi, men det afspejledes også i projektets mission om, at det skulle være tilladt at ødelægge softwaren for at

afprøve nye løsninger. Udover, at det skulle være sjovt, blev det også antaget, at personer deltog for at lære noget og udforske nye muligheder, som måske ikke ville have været muligt i andre projekter.

Der skal være plads til alle

Denne antagelse byggede på, at alle var velkommen, så længe de ville bidrage til projektet. For projektet betød det, at alle kunne deltage i de diskussioner, der fandt sted, samt at der var flere faglige niveauer, man kunne bidrage på.

Du skal bidrage

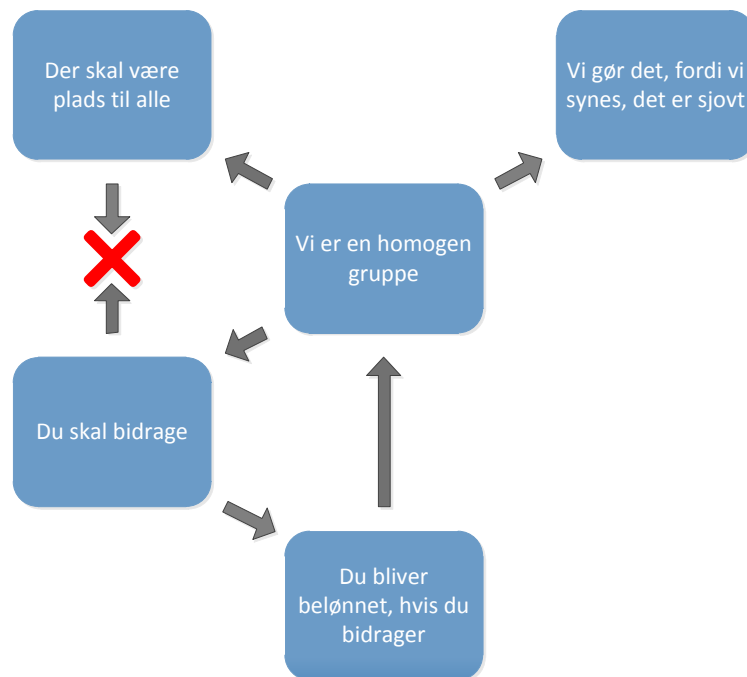
I projektet var der en antagelse om, at hvis man ønskede at få noget ud af communityet, skulle man give noget tilbage. Hvis man ikke bidrog, blev man anset som en belastning, og derved ville man møde et lukket community. Bidrag i forbindelse med denne antagelse skulle forstås bredt, det ville sige, at selvom du måske bidrog med kode, kunne du modarbejde projektet ved at repræsentere nogle andre værdier. Ikke blot med at man skulle deltage aktivt, så skulle man også bidrage til de fælles værdier, der herskede i projektet.

Du bliver belønnet, hvis du bidrager

Der var en antagelse i projektet, at hvis man gjorde en indsats og bidrog til projektet, skulle man også belønnes for sin indsats. Belønningen var i form af status relateret til den sociale rangstige.

12.2.4 Det kulturelle paradigme

Det kulturelle paradigme for Exherbo er skitseret i Figur 17. Paradigmet tog afsæt i antagelsen om, at de var en homogen gruppe, som delte en række værdier. Denne homogene gruppe antog, at man deltog, fordi man fandt projektet sjovt og lærerigt. Ydermere gav denne homogenitet anledning til to antagelser, hvor der kunne opstå en konflikt. Det var antaget, at der skulle være plads til alle, så længe de bidrog til projektet. Dette fungerede så længe medlemmer delte de samme værdier. Når en given person bidrog til projektet, blev personen belønnet ved at blive eksponeret overfor resten af gruppen, og derved ville personens status stige, da andre blev gjort opmærksom på personens bidrag til projektet. Dette kunne have en selvforstærkende effekt, hvor personen motiveredes yderligere og ville blive modtaget positivt, da personen ville blive anset som en del af den homogene gruppe. Problematikken opstod, når personer modarbejdede de værdier, der herskede i projektet. Så selvom en given person deltog aktivt i communityet og bidrog til projektet, kunne personen risikere at blive udelukket for projektet, hvorved der ikke længere var plads til alle, selvom de levede op til de umiddelbare kriterier for optagelse i gruppen.



Figur 17 Exherbo's kulturelle paradigme

12.3 CBPP processen

I og med at der var tale om en Linux distribution, betød det, at der var en række forskellige licenser, som de forskellige systemer, software osv. var udgivet under. Dette havde betydning for, hvordan distributionen kunne interagere med andre systemer. De dele, der blev udviklet i projektet, blev typisk udgivet under GNU GPL v2. Denne licens var dels en afspejling af BØs personlige holdning til OSS, men det havde desuden en praktisk fordel, da det gjorde processen med at interagere med andre OSS licenser nemmere. Det blev nemmere, da en række permissive licenser var kompatible med GNU GPL licensen. Denne kompatibilitet var envejs. Dvs. hvis man benyttede en restriktiv licens som GNU GPL, kunne softwaren ikke inkorporeres i andre software programmer uden denne software også blev udgivet under samme GNU GPL licens⁹². Projektet var derved beskyttet og tilgængeligt igennem de webbaserede teknologier, som projektet benyttede sig af, hvilket betød, at projektet havde alle forudsætningerne for at kunne drage nytte af ressourceallokeringsfordelen.

Det potentielle, der ligger i ressource allokeringfordelen, kan blive realiseret, når projektet udnytter de muligheder, der ligger i informationsbehandlingsfordelen, som bunder ud i en bred brugerdeltagelse og filtrering af kontributionerne til projektet. I dette projekt var der en bred brugerdeltagelse, hvoraf en mindre mængde var en del af kernegruppen. Projektets distribuerede udviklingsmodel, samt at ingen havde

⁹² For yderligere information omkring GNU GPL licensen henvises til GNU GPL officielle hjemmeside, den seneste version af GNU GPL licensen kan findes ved at navigere til <http://www.gnu.org/licenses/gpl.html>.

ejerskab over en given del af projektet, betød, at projektet nemt kunne drage fordel af et stort community. Projektet udviklingsmodel var baseret på en række opdelte VCS repositories efter emne. Denne modularitet gjorde det nemmere, samt at man kunne deltage med en relativt lille indsats, da udviklerne kunne vælge opgaver, som passede bedst i forhold til deres interesser og fagkundskaber⁹³. Kommunikationen i projektet foregik primært via IRC, hvilket betød, at asynkron kommunikation blev en udfordring for projektet. Det anbefalede, at medlemmerne lod deres IRC klient være på, selv når de ikke var fysisk tilstede for at kunne følge med i diskussionerne. Alternativt til IRC benyttede projektet desuden en mailingliste, som gjorde asynkron kommunikation nemmere. I forhold til filtrering benyttede projektet en IRC Bot, som stod for at holde styr på alle kontributioner til projektet, hvorefter det var op til communityet at læse koden igennem. Derefter var det op til kerneudviklerne at committe koden til projektets VCS.

Både ressourceallokerings- og informationsbehandlingsfordelen kræver, at man overkommer motivationsudfordringen. Projektet var drevet primært af glædesbaseret indre motivation, hvor medlemmer satte pris på, at det skulle være sjovt og gerne lærerigt. Ydermere var der indikationer på, at der var fællesskabsbaseret indre motivation tilstede i communityet, som afspejledes i kulturen, men datagrundlaget var ikke tilstrækkeligt til at fastslå dette. Indikationerne var bl.a., at man så vidt muligt blev belønnet for sine bidrag, specielt når man var nyttilkommen, hvilket skabte en tilknytning til resten af communityet, samt at udviklerne generelt blev i projektet i en længere periode. Projektets udviklingsmodel gjorde det muligt at udnytte medlemmernes forskellige motivationer for at deltage i projektet. Udviklingsmodellen egnede sig desuden til at kunne drage nytte af skalerbar motivation, da projektet var delt op i mindre emner, som udviklerne kunne vælge at arbejde på.

Når ressource allokerings- og informationsbehandlingsfordelene er håndteret, opstår der et behov for koordineret handling, som er en del af organisationsudfordringen. Som beskrevet i afsnittet om arbejdsdeling blev projektet koordineret decentralt, hvor de overordnede rammer blev sat af kernegruppen⁹⁴. Opgaverne blev stillet både af kernegruppen og det aktive community. Det var dog stadig kernegruppen, som havde det overordnede ansvar og kunne fravælge opgaver, hvis de ikke så det som værende væsentlig nok til at blive en del af standardkonfigurationen. Når opgaverne var definerede var det op til den enkelte at vælge, hvilken opgave personen havde lyst til at arbejde på. Dette gav mulighed for, at der arbejdedes på flere løsninger til samme opgave.

Efter at opgaverne i projektet var koordinerede, opstod behovet for integration. I projektet var der en demokratisk tilgang til, hvilken løsning der skulle anvendes. Hvis en løsning ikke kunne vælges, var det op til

⁹³ (Andreev, Feller, Finnegan, & Moretz, 2010, s. 7)

⁹⁴ BØ er også en del af kernegruppen, selvom han har fået tildelt sin egen rolle.

kernegruppen at vælge en løsning. Hvis det endnu ikke var muligt at vælge, var det BØ, som tog den afgørende beslutning om, hvilken løsning der skulle anvendes. Det var desuden kernegruppens opgave at beslutte, om en given løsning skulle integreres i Exherbo's VCS eller alternativt, om der skulle skabes en relation via deres pakkehåndteringssoftware til et eksternt VCS, som ikke ville blive varetaget af Exherbo projektet.

12.4 Kritiske valg

De identificerede kritiske valg i projektet har primært handlet om at skabe et community med en kultur, som bestod af en homogen gruppe, der delte en række bestemte værdier.

12.4.1 K1 - Kulturelle værdier

Da det gik op for BØ, at han havde startet et nyt projekt, valgte han fra starten af at definere en række kulturelle værdier, som reflekterede, hvad han anså for ønskværdigt og uønskværdigt. Dette valg var en afledt effekt af hans tidligere arbejde i Gentoo projektet, hvor han oplevede en kultur, han ikke kunne stå inde for. Når der kom nye til projektet, gjorde han meget ud af, at personerne skulle opleve, hvilke værdier han så som ønskværdigt. Bl.a. gjorde han meget ud af, at nytilkommende skulle have en hurtig succesoplevelse, om det var et hurtigt commit af en patch, eller om han skulle bruge et par timer på at hjælpe en person med at rette en trivial fejl. Han gjorde, hvad der skulle til for at give dem den oplevelse. Dette engagement havde en direkte påvirkning på den motivation, man kunne finde i projektet, hvor flere udviklere blev i communityet over en længere periode. Den glædesbaserede motivation var den type, som direkte blev påvirket af denne kultur, da de fik noget ud af at bidrage til projektet. En afledt motivation var den fællesskabsbaserede motivation, der blev styrket, da BØ kunne observere, at kerneudviklerne videreførte projektet med samme værdier.

Dette valg har været et essentielt valg for projektet, da det definerede projektet og dets kultur. I forhold til CBPP processen udover motivation som nævnt ovenfor har det haft en direkte påvirkning af, hvordan projektet overkom de organisatoriske udfordringer. Reelt havde lederne ingen magt over de andre medlemmer, da de deltog på deres egne præmisser, men ved at have en et værdisæt, var det muligt at styre organisationen i en bestemt retning. Det var derfor muligt at undgå en række konflikter, da det var muligt at identificere afvigere og minimere deres negative indflydelse på projektet.

12.4.2 K2 - Overgang til git

Git har bidraget til belønningssystemet. Tidligere benyttede projektet SVN⁹⁵, hvilket ikke direkte understøttede denne form for belønningssystem. Ved at bruge git var det muligt via en standardiseret proces at at-

⁹⁵ SVN er et VCS.

tribuere et commit til en person, som ikke havde commit adgang. Dette gjorde det væsentligt nemmere for kernegruppen at håndtere kontributioner fra andre uden at skulle tænke på, at de selv skulle indsætte navnet i fx kommentarfeltet. Dette var essentielt, da værdierne i projektet foreskrev at, man skulle belønnes for sin indsats. En af måderne var at blive krediteret for sit bidrag, som dette valg gjorde muligt.

Valget at benytte git var mindre væsentligt for projektet, men det var kritisk i den forstand, at det har været med til at understøtte projektets værdier samt automatisere en proces som ellers kunne være påvirket af menneskelige fejl.

12.4.3 K3 - Manifest

Manifestet, som lå på forsiden af projektets hjemmeside, fik en betydning for projektet, da det var med til at signalere, hvilken type projekt det var samt deres syn på omverdenen. Manifestet gav et billede af et projekt, som adskilte sig betydeligt fra, hvordan projektet er i skrevne stund. Manifestet opfordrede folk til at blive væk og lade være med at kontakte dem. Manifestet beskrev bl.a., hvorfor projektet begyndte. I forbindelse med dette blev Gentoo projektet kritiseret⁹⁶, og det har givetvis været med til at opildne en konflikt imellem Gentoo projektet og Exherbo.

Ved at fjerne manifestet betød det et brud med, hvordan projektet anså omverdenen, hvilket dannede grobund for den type organisationen, projektet er i skrevne stund. Samtidig betød det også en ændring af, hvordan projektet blev opfattet af omverdenen. Mens manifestet var en del af hjemmesiden, blev signalerne modtaget tvetydigt, da nogle personer troede, det var et forsøg på at skjule et genialt projekt. Ved at fjerne manifestet var det med til at skabe et ensartet billede af, hvilken type projekt det var.

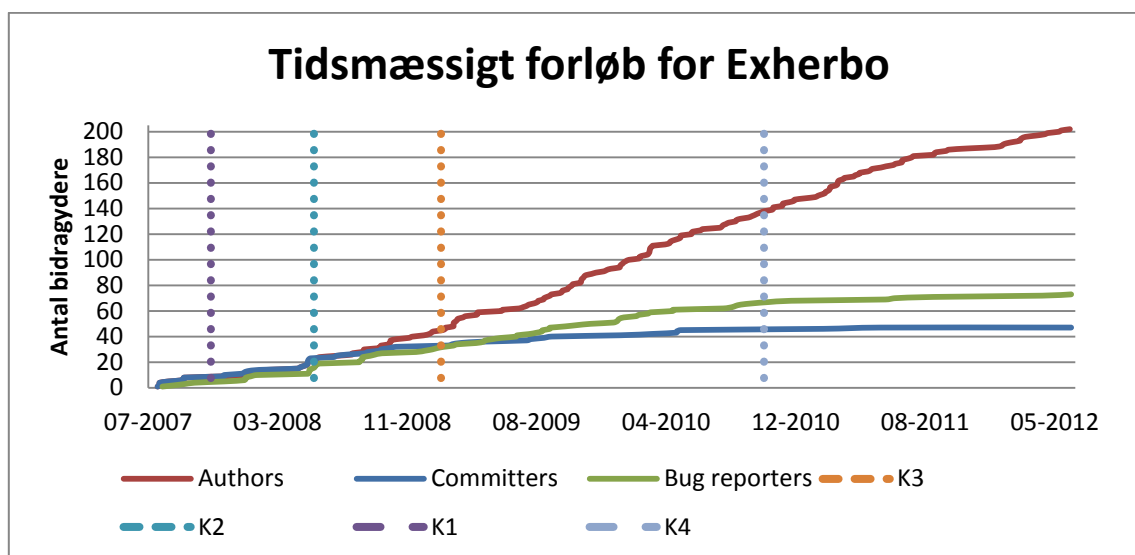
12.4.4 K4 - Forventninger

Inden der blev skrevet på projektets hjemmeside, hvilke forventninger der var til brugerne, oplevede medlemmerne, at der kom brugere til, som havde nogle helt andre forventninger til projektet, end der var gældende. Dette resulterede i nogle fjendtlige konfrontationer blandt de involverede parter. Ved at formulere en række forventninger, var det muligt på forhånd at frasortere en mængde, som ikke hørte til den homogene gruppekultur, der var opbygget i projektet. Ved at kunne håndtere mulige konflikter før de skete og uden direkte indblanding fra medlemmerne, har det været med til at søge for et mindre konfliktfyldt projekt. Dette kunne have en positiv indirekte effekt på medlemmernes motivation, da færre konflikter var med til at styrke kulturen og gøre det mere behageligt for medlemmerne at deltage i projektet.

⁹⁶ Se Bilag 14 for et uddrag af manifestet hvori Gentoo bliver genstand for kritik.

12.4.5 Kritiske valg sammenfattet

Som det fremgår af Figur 18 er der indikation på, at de kritiske valg⁹⁷ har været med til, at projektet har kunnet fastholde en konstant vækst i projektets løbetid. Det interessante ved dette projekt har været forskellen på mængden af authors og bug reporters. Størstedelen valgte at deltage ved at kontribuere kode til projektet. Dette understøttede de værdier, der fandtes i projektet, hvilket indikerede, at kulturen havde en afgørende effekt på, hvilke personer der involverer sig i projektet.



Figur 18 Oversigt over kritiske valg set i forhold til projektets vækst. De prikkede linjer angiver, hvornår et kritisk valg er opstået eller realiseret og skal læses i forbindelse med y-aksen. De solide streger viser udviklingen over tid for henholdsvis committers, authors og bug reporters. De skal læses i forhold til den venstre y-akse.^{98 99}

13 Konfigurationer

I forbindelse med identifikationen af kritiske valg har det resulteret i syv generiske kritiske valgkategorier, som de enkelte valg blev inddelt i. De syv kategorier er licensvalg, arkitektur, distributioner, ledelse, kultur, omverdenen og projektstop. I Tabel 2 er de kritiske valg opsummeret og sat i relation til en hovedkategori.

Kritisk valg	Bromine	Varnish Cache	Exherbo
Licensvalg	GPL	FreeBSD Konflikt imellem medlemmer	

⁹⁷ Se Bilag 23 for en oversigt over Exherbos kritiske valg.

⁹⁸ Bilag 12 indeholder en oversigt over, hvilke git repositories der er blevet brugt til udregning af authors og committers.

⁹⁹ Det samlede overblik over projektet, som er skitseret i grafen indeholder ikke informationer omkring den teknologiske udvikling. Dette skyldes, at der ikke er fundet data, som kan repræsentere udviklingen. Projektet benytter ikke versionering i almindelig forstand, fx version 1, 2 osv.

Arkitektur	Version 1-2 CakePHP	Kode hygiejne VMODs	
Distributioner	Self-installer VM	Nemt at installere	
Ledelse	Autonom ledelse Lederskift	Bestyrelse	Bevidst omkring projektets værdier Åbenhed omkring forventninger
Kultur	Autonom ledelse	Vi skal være de bedste	Bevidst omkring projektets værdier Skift til git Manifest Åbenhed omkring forventninger
Omverdenen	Optagelse ved OpenQA	VUG	
Stop projekt	Mangle på motivation		

Tabel 2 Kritiske valg oversigt

At et projekt ikke har taget et kritisk valg, betyder ikke, at projektet ikke har taget et valg inden for det pågældende område men blot, at det ikke er bedømt som værende af kritisk karakter for projektet. Fx har Exherbo ikke taget et kritisk valg i forbindelse med licensvalg. I stedet har licensvalget været en refleksion af udviklernes personlige præferencer samt af hensynet til at gøre arbejdet nemmere med andre GNU GPL licenser.

Der gives bud på klassificering af konfigurationer. Disse bud er dog med forbehold, da hvert projekt har en unik konfiguration af kritiske valg, hvor et større datagrundlag vil være påkrævet for at kunne konkludere på et mere generelt plan med udgangspunkt i de primære drivere vedrørende motivation, kultur, organisation og softwaretype.

Parametrene er valgt ud fra CBPP processen. Dette parameter danner rammerne for et projekt. Der er inddraget, hvilken type software projekter udvikler, for at vise typer af software som fx en Linux distribution, der har et langt større potentiale, når det kommer til kerneudviklere, hvorimod et projekt som Varnish er væsentlig begrænset i forhold til, hvor mange udviklere der er plads til.

Projekt	Motivation	Kultur	Organisation	Software type
Bromine	Ydre motivation	Alle er lige Ingen ledelse	Hybridorganisation	Webbaseret platform
Varnish Cache	Ydre motivation	Udsat projekt Stiller høje krav	Katedralorganisation	HTTP accelerator

Exherbo	Indre motivation	Man skal bidrage	Basarorganisation	Linux distribution
		Homogen kerne- gruppe		

Tabel 3 Konfigurationer

Disse konfigurationer vil kunne bruges ved en større undersøgelse som inddelingsparametre, som vil kunne give et mere entydigt syn på, hvilke typiske valgkonfigurationer, der findes inden for OSS projekter.

14 Generalisering af kritiske valg

Til generalisering af de kritiske valg er der taget udgangspunkt i de syv kritiske valgkategorier, som blev defineret i afsnit 13. Disse generiske valg er sat i en bredere kontekst, hvor målet har været at give en vejledning til OSS iværksættere. Vejledning består i eksempler fra de tre projekter, hvor mulige fordele og ulemper ved et given valg er identificeret.

14.1 Licensvalg

Når beslutningen om at et given projekt skal udgives som open source, stilles projektmedlemmerne over for spørgsmålet om, hvilken licens der skal benyttes¹⁰⁰.

Valg af en restriktiv licens kan være en hæmsko for et projekts udbredelse pga. licensens natur. Fx kan en restriktiv licens have en politisk agenda, som kan udmøntes i de rettigheder, man har opnået ved at benytte denne licens, hvor disse skal videregives til alle andre. Dette kan være et problem i kommerciel henseende. Det kan til gengæld være en fordel for et projekt at vælge en restriktiv licens, hvis man selv deler denne ideologi.

Valg af en permissiv licens kan være en fordel for et projekt, hvis man blot ønsker en så vid udbredelse som mulig. Det betyder dog også, at alle er lige, hvilket som ophavsmand kan betyde en øget konkurrence, især hvis man har et ønske om at kunne drive forretning på baggrund af projektet.

Uanset hvilken licens man vælger at benytte, vil det få konsekvenser, når projektet vokser og flere personer bliver tilknyttet, da alle kontributioner vil være bundet af den valgte licens, såfremt den enkelte person ikke vælger, at hans kontributioner skal udgives med en anden licens. Hvis det er tilfældet, at en person ikke ønsker at benytte samme licens som projektet, kan det medføre en række komplikationer, hvor det kan være, at man er bedre tjent uden denne persons kontributioner. Ophavsretten vil under alle omstændigheder tilhøre den person, som har afgivet fx et stykke kode. Hvis man på et senere tidspunkt ønsker at gøre noget med projektet, som er uforeneligt med den valgte licens, vil man skulle fjerne alle de kontributioner,

¹⁰⁰ Se afsnittet 2.1 hvor de forskellige licenstyper defineres.

hvor man ikke har kunnet få samtykke fra ophavsmanden. Det samme ville gøre sig gældende, hvis man ønsker at ændre licens undervejs. Enten får man samtykke fra samtlige personer som har bidraget til projektet eller også skal man fjerne de dele, der ikke kunne indhentes samtykke omkring.

Når man vælger en licens til sit projekt, og man på daværende tidspunkt ved eller overvejer at inkorporere andre open source projekter, vil man skulle tage højde for disse projekters licens. Det er ikke alle licenser, der er kompatible med hinanden.

I sidste instans afhænger valget af licens af, hvad man ønsker at opnå med projektet samt, hvad man selv kan stå inde for. Er man mere interesseret i de ideologiske aspekter af open source, vil en restriktiv licens være en mulig kandidat, hvorimod hvis man blot ønsker, at andre skal bruge det, samt øge anvendelsesmulighederne, vil en permissiv licens være en mulig kandidat. Om end er det vigtigt, at man tager et valg og så vidt forsøger at holde fast i det valg, da licensvalget næppe vil afholde folk fra at bruge eller bidrage til projektet. Hvis det er tilfældet, at licensen er et problem for nogle, vil man formentlig være bedre tjent uden disse personer.

14.2 Arkitektur

Når det kommer til valg af arkitektur, er der mange muligheder. Mange valg vil kunne reduceres til datalogiske problemstillinger. De valg, som er blevet betegnet som kritiske i denne afhandling, har været tilknyttet to områder henholdsvis kodehygiejne og support af plugins.

Kodehygiejne har den effekt, at det sørger for, at der er styr på kodebasen både testsmæssigt men også, at der ikke er kode, som ingen kender funktionen af. Kodehygiejne er et parameter, man kan skrue på alt afhængig af ens personlige temperament. Under interview med RBP kom det frem, at han havde deltaget i et andet OSS projekt end Bromine, hvor projektejereren havde valgt en så stram teststrategi, at RBP mistede sin motivation til at samarbejde med projektet. I Bromine blev det krævet at al kode, der blev bidraget skulle indeholde tests cases, som dokumenterede, at patchen virkede. I Bromine var der en meget løs teststrategi, hvor udviklerne blot testede, om en patch virkede og dernæst committede den. Disse to projekter kan anses for ekstremer, hvor en havde en meget stram teststrategi modsat Bromine hvor der stort set ingen teststrategi var. Hvor man lægger sig, kan influere på, hvilke personer man får med i sit projekt og derved også, hvem man fravælger. At finde et sted imellem disse to ekstremer vil være at foretrække, og afhængig af hvor kritisk et system er, vil indikere, om man skal have meget fokus på, at den kode, der bliver indgivet, er gennemtestet og veldokumenteret.

14.3 Distributioner

Afhængig af hvilken type bruger man henvender sig til, vil valget af distributioner være en afgørende faktor. I Bromine blev den typiske bruger anset som værende ikke-teknisk. Derfor blev det valgt at udgive softwaren, som ville gøre det nemmere for brugeren at komme i gang med at bruge softwaren. Bromine skal ses i det lys, at de afhang af en række tredjepartsprogrammer, som krævede en teknisk viden.

Dette var dog ikke den eneste grund til, at man skal udgive distributioner, som gør det nemmere for brugeren. I Varnish oplevede de, at flere henvendte sig til projektet, når deres hjemmeside havde et problem, som Varnish var designet til at løse. Ved at give dem mulighed for en hurtig opsætning via de indbyggede pakkehåndteringssystemer, som fandtes i fx Ubuntu, hjalp det brugerne med at komme i gang med at bruge Varnish. Dette skabte en stor værdi for brugeren. Ydermere flyttede denne type pakkehåndteringssystemer en del af udviklingen af installationsmedier over til andre, hvorved man slap for selv at udvikle et system med en lignende funktion.

Selvom ovenstående taler for, at man skal gøre det så nemt for brugeren som muligt, er det vigtigt ikke at glemme, at nogle brugere også selv vil stå for installationen og konfigurationen, hvor et VCS spiller en central rolle.

Det gælder derfor om at identificere sine brugere og give dem de muligheder, de har brug for. Hvis communityet er meget tekniske og egentlig er mest interesserede i selv at stå for det hele, vil det være spild af tid at genere en række virtual appliances. Hvis de alternativt mangler tekniske færdigheder, vil et virtual appliance være en oplagt mulighed.

14.4 Ledelse

I de tre udvalgte projekter blev der benyttet tre forskellige ledelsesstile. Alle tre ledelsesformer havde deres fordele og ulemper.

I Bromine herskede en betydningsfuld grundlæggende antagelse om, at man ikke kan tvinge folk til at deltage. Ideen om ikke at tvinge folk var korrekt, men problemet i Bromine var, at det ikke blev håndteret, hvor i Exherbo havde man brugt kulturen som et middel til at motivere folk. Når man taler om udgivelser, bliver denne antagelse relevant. PHK har forklaret, at han har oplevet i flere projekter, at man har den antagelse, at man både kan sætte en deadline for, hvornår en udgivelse skal falde samt, hvilke funktioner den skal indeholde. Denne antagelse har et problem, da man som udgangspunkt ikke kan tvinge folk til at arbejde på projektet. Derfor kan man reelt kun vælge et af de to parametre. Enten sætter man en udgivelsesdato, og så kommer de funktioner med, som er færdige, eller også vælger man, hvilke funktioner der skal

med, og når funktionerne er færdige bliver softwaren udgivet. Hvis man vælger at benytte begge parametre, kan det ende ud i en konflikt. Som i sidste instans kan skade projektet som hel hed.

14.5 Kultur

Kultur kan være essentielt for et projekt. Ved aktivt at tage stilling til kulturen kan det være med til at løse flere udfordringer for et projekt. I Exherbo ses det tydeligt. Ved at forventningerne afstemmes på forhånd, har det været muligt at skabe et community som har dele mange af de samme værdier. Når et community deler de samme værdier vil der være et betydeligt mindre behov for at lederne aktivt skal lede projektet. Indsatsen lægges tidligt i processen, når folk henvender sig til projektet for at deltage aktivt. Ved at de nytilkommende bliver indoktrineret i kulturen fra start af, betyder det, at de tager værdierne til sig hurtigt og derved vil være i stand til at viderebringe dem, når endnu en ny kommer til projektet. En måde hvorpå Exherbo har gjort dette, er ved at give nytilkommende nogle hurtige succes oplevelser.

Lige gyldig hvilke værdier man ønsker at fremme i et projekt, vil det være essentielt, at de nytilkommende får nogle hurtige succesoplevelser. Dette vil øge sandsynligheden for, at de bliver i projektet i en længere periode. Bromine er et fint eksempel på det modsatte, hvor ingen udover de originale tre udviklere blev i projektet over en længere periode. Dette kunne relateres direkte tilbage til kulturen og ledelsen af projektet.

14.6 Omverdenen

Omverdenen skal ses som en mulighed. Det kan være en fordel at skabe kontakt til andre communities. Dette kan være med til at promovere projektet samt kan være med til at kickstarte projektets community. I den forbindelse skal man have gjort nogle tanker omkring, hvad man forventer af et community, da man ellers risikerer, at folk hurtigt mister interessen for projektet.

Ved at se på de muligheder, der ligger uden for de traditionelle rammer for et OSS projekt, kan man skabe et forum, som vil sætte folk sammen, som ellers ikke ville have været muligt. I Varnish projektet har man valgt at benytte brugermøder. Disse møder skaber en mere uformel kontakt bland communityets medlemmer, som kan være med til at fremme et samarbejde, som ellers ikke ville have været muligt. Hvis man har god kontakt til sit community kan en brugergruppe selv med et mindre community være en god ide. Fx var der kun 14 deltagere¹⁰¹ ved Varnish første møde.

Det vil derfor være en fordel, hvis et projekt så på dets omgivelser, for at identificere mulige samarbejdspartnere og undersøge, hvilke muligheder der ligger i omgivelserne, som projektet kan drage nytte af.

¹⁰¹ Se Bilag 20

14.7 Stop projektet

Det kan være nødvendigt at stoppe et projekt. I Bromine projektet var det afgørende argument den manglende motivation. Hvis man skal undgå en lignende situation, er det vigtigt, at man forholder sig til sit community, da det vil være nogle fra communityet, som vil skulle videreføre projektet. Kulturen vil spille en central rolle i at fastholde personer fra communityet, som beskrevet i afsnit 14.5.

15 Konklusion

I dette afsnit vil problemformulering blive besvaret.

Hvad motiverede opstarten af projektet?

Exherbo projektet udsprang fra et personligt behov hos BØ efter hans brud med Gentoo projektet. Exherbo var primært drevet af glædesbaseret motivation, hvor det tekniske var det interessante. Da projektet begyndte at vokse, kom der indikationer på, at en fællesskabsbaseret motivation var til stede. Denne motivation kom til udtryk i forhold til de værdier der var repræsenteret i projektets kultur.

Både Bromine og Varnish udsprang fra en kommerciel kontekst, hvilket også blev afbilledet i deres motivation. Begge projekter var primært drevet af ydre motivation, hvor monetær godtgørelse spillede en væsentlig rolle. For Bromine fik det senere konsekvenser for projektet, da udviklernes engagement faldt, idet de valgte at stoppe på Eniro. Varnish vil komme til at stå overfor en lignende udfordring, når PHKs VML ikke længere er nok til at holde ham beskæftiget på projektet.

Hvordan er projektet organiseret?

Alle projekterne havde en forskellig tilgang til organisationsstrukturen. Exherbo var organiseret som en basarorganisation, Varnish var centraliseret og lignede til dels katedralorganisationen, og Bromine kunne anses som en hybridorganisation imellem basar- og katedralorganisationen.

Kulturen i Exherbo byggede på en homogen kernegruppe, som delte en række værdier. Man skulle bl.a. bidrage til fællesskabet, som resulterede i en belønning i form af status. Projektet havde en antagelse omkring, at der skulle være plads til alle, så længe de bidrog til projektet. Der var dog en konflikt mellem at bidrage samtidig med, at der skulle være plads til alle. Hvis man modarbejdede den homogene gruppes værdier, kunne man risikere at blive ekskluderet, også selvom man bidrog til projektet. Generelt var kulturen et sted, hvor man skulle have det sjovt, og hvor der blev lagt vægt på læring. Kulturen blev desuden brugt som ledelsesredskab.

Kulturen i Varnish var præget af et ønske om at være de bedste, hvilket var med til at skabe en kultur, som gjorde det svært at blive en del af kernegruppen. Kulturen var med til at holde kravene oppe. Kravene ud-

sprang fra softwarens kompleksitet. Dette betød, at projektmedlemmerne antog, at man hellere ville gøre tingene langsomt og rigtigt frem for hurtigt og forkert. Dette var med til at sætte kulturen i en udsat position, da projektet primært af var drevet af en udvikler. Kulturen kunne derfor forklares som værende af en cirkulær form med en selvforstærkende effekt. Ledelsen af Varnish var en refleksion af kulturen, da projektet blev drevet af et fåtal, betød det også en meget centraliseret ledelse.

Kulturen i Bromine var kendetegnet ved at projektet var godt for fællesskabet. Udviklerne brugte tid i og uden for deres arbejde på projektet. Kulturen var præget af, at alle blev anset som ligeværdige, og man ikke kunne tvinge andre til at deltage aktivt. Denne grundlæggende antagelse, om at alle var lige, blev cementeret i ledelsen. Ledelsen kunne karakteriseres som autonom eller helt fraværende efter bruddet med Eniro. Efterhånden som projektet voksede, følte udviklerne et ansvar overfor communityet. Dette ansvar var med til at holde projektet i live efter, de forlod Eniro. Efter bruddet viste en latent grundlæggende antagelse sig, som handlede om, at når man ikke brugte softwaren, var projektet ikke interessant. Dette resulterede i, at projektet blev nedlagt ca. 1½ år efter deres brud med Eniro.

Hvordan påvirker den teknologiske udvikling af softwaren projektet?

Et valg, der direkte kunne relateres til den teknologiske udvikling, var da Varnish og Bromine valgte at åbne arkitekturen op ved at implementere et interface som understøttede plugins. Om det var et tilfælde, at begge projekter gjorde det i version tre, eller om det var en indikation på et mere universelt resultat af softwarens kontinuerlige udvikling, var datagrundlaget ikke tilstrækkeligt til at konkludere. Det man kunne konkludere, var, at projekterne havde været igennem en modningsproces, som resulterede i, at version tre var det tidspunkt, hvor arkitekturen blev åbnet op. Exherbo projektet adskilte sig fra de to andre projekter ved, at det var en Linux distribution, og det var derfor ikke muligt at tale om plugins i samme forstand.

I forhold til forventninger gjorde Exherbo projektet meget for at skildre, med hvilke forventninger de havde til brugerne, og hvad de kunne forvente at få igen. Oprindeligt havde projektet et manifest, hvor de forsøgte at afskrække folk fra at bruge det, da de ikke selv mente, at det var klar til offentligheden endnu. Hverken Bromine eller Varnish forsøgte at gøre noget lignende. I stedet havde det være op til folk at komme til projekterne, og dernæst kunne forventningerne afstemmes. Dette resulterede i, at personer kun kort involverede sig i projekterne.

Generelt når det kommer til vækst i et OSS projekt, vil kulturen have en afgørende effekt på, hvordan væksten bliver håndteret i projektet. I Exherbo projektet blev kulturen brugt som et styringsredskab, som var med til at håndtere nytilkommende, men samtidig kunne kulturen også bruges som et magtmiddel til at ekskludere folk, når der begyndte at opstå konflikter i projektet. I Bromine projektet blev vækst ikke håndte-

ret. Dette afspejledes i det kulturelle paradigme, hvor at ingen tog det overordnede ansvar, da alle blev anset som værende ligeværdige. I Varnish sad PHK hårdt på projektet, hvilket betød at kernegruppen havde en minimal vækst. Dvs., der ikke var nogen betydelig indflydelse på projektet. Igen var dette en refleksion af det kulturelle paradigme.

Hvordan påvirker følgende kritiske valg projektet?

Ikke alle kritiske valg kunne betegnes som kritiske for det enkelte projekt. De valg, som blev betegnet for kritiske, påvirkede projektet i forhold til struktur, kultur, motivation eller CBPP processen.

Licensvalg påvirkede projekterne i forhold til motivation og en bred brugerdeltagelse. Hvis et projekt blev udarbejdet i forbindelse med en virksomhed, kunne en open source licens være med til, at udviklerne fik ejerfølelser over projektet, som gjorde, at de kunne være villige til at bruge deres egen tid på udvikling. Licensvalg kunne også skabe konflikt. I Varnish gav det en konflikt imellem APJ og Varnish Software, da Varnish bogen blev udgivet under en restriktiv licens, hvoraf selve projektet var udgivet under en permissiv licens. Ligeledes ved at benytte en permissive licens åbnedes muligheden for en bredere brugerdeltagelse.

Arkitekturen havde et potentiale til at øge en bred brugerdeltagelse. Både i Bromine og Varnish forsøgte man sig med at lave en arkitektur, der understøttede plugins. Bromine havde dog ikke succes med dette, men i Varnish projektet var der indikationer på, at det kunne blive en succes.

Distributioner spillede en vigtig rolle i at nå ud til et projekts målgruppe. Både Bromine og Varnish forsøgte at gøre det så nemt for deres brugere som muligt ved at distribuere komplette pakker, som ikke krævede opsætning. Distributioner kunne være med til at afspejle projektets værdier.

Ledelse i et projekt kan have mange udfald. I Bromine betød en mangel på lederskab, at projektet ikke var i stand til at fastholde nytilkommende, da der ikke blev taget hånd om dem. I Varnish blev projektet styret centralt, hvilket var en refleksion af, at PHK var den primære drivkraft. Dette resulterede i, at projektet var meget udsat, for hvis PHK forlod projektet, var der reelt ikke nogen, der var klar til at overtage. I Exherbo blev kulturen brugt som ledelse. Igennem en række definerede værdier blev kerneudviklere indoktrineret i projektets kultur. Dette resulterede i en bred brugerdeltagelse samt en homogen kernegruppe.

Kulturen spiller en central rolle i et OSS projekt. Kulturen kan være med til at fastholde folk i communityet, som var tilfældet i Exherbo, men det kan også være med til, at personer kun kort involverer sig som det var tilfældet i Bromine og Varnish.

Omverdenen kan bruges til at fremme et projekt, som Bromine gjorde ved at søge om optagelse i et vel-etableret community. For at inddrage omverdenen havde Varnish benyttet sig af brugergrupper, hvilket var

med til fremme communityet, da det gav en mulighed for at mødes, som e-kommunikation ikke var egnet til. Omverdenen kunne også resultere i konflikter. I Exherbo var der en interprojektkonflikt med Gentoo projektet. Hvor kritiske disse konflikter var, fandtes der ikke belæg til at konkludere i denne afhandling. Det som kunne konstateres var, at der opstod et forventningsproblem imellem nogle bruger og Exherbo projektet.

At stoppe et projekt er det ultimative valg, man kan tage. I Bromine projektet blev dette valg taget på baggrund af en manglende motivation fra kernegruppen. Det blev forsøgt at videregive projektet, men da communityet ikke blev styret, var der ikke nogen til at overtage hele projektet.

Hvilke typiske kritiske valg konfigurationer findes der i OSS projekter?

I denne afhandling blev tre forskellige konfiguration identificeret. Konfigurationerne blev udspecificeret ved at se på følgende parametre: motivation, kultur, organisation samt typen af software, der blev udviklet.

Det har derfor ikke været muligt at give et bud på, hvilke typiske valgkonfigurationer, der findes i OSS projekter, da datagrundlaget ikke var tilstrækkeligt. I stedet kan resultatet bruges til udvælgelse af en større population og dernæst sammenligne, hvilke kritiske valgkonfigurationer, der findes i samme konfiguration, som er identificeret i denne afhandling.

Kan OSS projekternes kritiske valg og deres løsninger anvendes til vejledning af iværksættere (generaliseres til iværksætter teori)?

Ved at generalisere de kritiske valg har det været muligt at anvende valgene til vejledning af OSS iværksættere. Ved at fokusere på en generalisering har det været muligt at se valgene i et større perspektiv, som viser, hvilke fordele og ulemper, der kan forekomme, når der skal tages et kritisk valg.

16 Diskussion

I analysen af projekterne kunne det være interessant at få inddraget den aktive del af communityet. Ved at inddrage communityet vil det være muligt at sætte de kritiske valg i et andet perspektiv. Ved at tage udgangspunkt i et andet perspektiv kunne det være med til at identificere nye kritiske valg, som ikke kunne frembringes i denne afhandling. Dette skyldes, at vægten i denne afhandling er lagt på kernegruppen, men det antages, at kernegruppen ikke er det eneste omdrejningspunkt i projektet, hvorved kritiske valg har et potentiale for at dannes i periferien af projektet.

Denne afhandling har taget udgangspunkt i nogle relative små projekter. Dette rejser spørgsmålet, om der er forskel på kritiske valg i henholdsvis små og større projekter. Der var en potentiel mulighed for at undersøge dette i Exherbo projektet, men da der kun blev foretaget interview med et nøglemedlem, har data-

grundlaget ikke været tilstrækkeligt til at konkludere på baggrund af størrelsen af et projekt. Dette var tilfældet, selvom personen var med fra starten af projektet og besad en stor mængde viden omkring communityet og projektet. Det ville have været interessant at inddrage andre kerneudviklere samt et udsnit af det aktive community. Disse kunne inddrages for at opnå en større forståelse af, hvilken rolle kulturen havde på projektet, og om der kunne findes en direkte relation til medlemmernes motivation.

De generiske kritiske valg, der blev identificeret og derefter undersøgt i de enkelte projekter, resulterede i en række specifikke kritiske valg. Disse specifikke kritiske valg kan ikke anses som en udtømmende liste af kritiske valg men derimod, at der ikke kunne identificeres andre kritiske valg i projekterne på undersøgelsestidspunktet. Andre kritiske valg, som ikke blev identificeret i denne afhandling, kunne være opkøb af rettigheder til OSS projektet eller hvis projektet skal forkes.

For fremtidige undersøgelser kunne det være relevant at finde projekter, som havde stået overfor ovenstående potentielle kritiske valg, da det ville være med til at skabe en bedre forståelse af open source og de udfordringer, der kan opstå i forbindelse med, at der skal tages et kritisk valg. Dette være sig gældende for både små og større projekter. Ydermere kunne det være relevant at inddrage, hvilken type software der er blevet udviklet i projektet. Datagrundlaget i denne afhandling omfattede ikke et tilstrækkeligt antal projekter for at kunne benytte type som en faktor. Typen kunne være relevant at inddrage, da en Linux distribution som udgangspunkt ville have et større potentiale i forhold til communityets størrelse set i forhold til Bromine og Varnish, som var afgrænsede slutbruger softwareprojekter.

17 Referencer

200909UserGroupMeeting – Varnish:. (u.d.). Hentede 29. maj 2012 fra <https://www.varnish-cache.org/trac/wiki/200909UserGroupMeeting>

Andersen, I. (2005). *Den skinbarlige virkelighed* (3 udg.). Forlaget Samfundslitteratur.

Andreev, P., Feller, J., Finnegan, P., & Moretz, J. (1. Januar 2010). Conceptualizing The Commonsbased Peerproduction Of Software: An Activity Theoretic Analysis. *International Conference on Information Systems*.

Apache Software Foundation. (u.d.). Hentede 11. juni 2012 fra <http://www.apache.org/foundation/how-it-works.html>

Buch-Hansen, H., & Nielsen, P. (2005). *Kritisk realisme* (1. udgave, 3 oplag 2008 udg.). Roksilde Universitetsforlag.

Jeppe Poss (2012). Open source: Kritiske valg

Dahan, J. (u.d.). *Features of Exherbo*. Hentede 10. juni 2012 fra <http://exherbo.org/docs/features.html>

Exherbo git repositories. (u.d.). Hentede 12. 6 2012 fra <http://git.exherbo.org/>

Exherbo Manifest. (u.d.). Hentede 14. juni 2012 fra <http://git.exherbo.org/docs.git/plain/manifesto/manifesto.txt?id=620c9ef8eb612b0d684504258d051424ce1cbf60>

Feller, J., Fitzgerald, B., Hissam, S. A., & Lakhani, K. R. (2005). *Perspectives on Free and Open Source Software*. Massachusetts Institute of Technology.

Fleetwood, S., & Ackroyd, S. (2004). *Critical Realist Applications in Organisation and Management Studies*. Routledge.

Garzarelli, G., Limam, Y. R., & Bjørn, T. (2008). Open Source Software and Economic Growth: A Classical Division of Labor Perspective. *Information Technology for Development*, 14 (2), s. 116-135.

Guba, E. G. (1990). *The Paradigme dialog*. Sage Publications, Inc.

Jacobsen, D. I., & Thorsvik, J. (2002). *Hvordan organisationer fungerer - Indføring i organisation og ledelse* (1 udg.). Hans Reitzels Forlag.

Nygaard, C. (2005). *Samfundsvidenskabelige analysemetoder* (1 udg.). Forlaget Samfundslitteratur.

Palm, R., Kløft, V., & Poss, J. (u.d.). Hentede 29. februar 2012 fra Bromine SVN - Assembla: <http://www.assembla.com/code/Bromine/subversion/nodes/trunk/README?rev=22>

Schultz, M. (1990). *Kultur i organisationer* (1. udgave, 5. oplag 1999 udg.). Handelshøjskolens Forlag.

Smith, B. (u.d.). *A Quick Guide to GPLv3 - GNU Project - Free Software Foundation (FSF)*. Hentede 5. juni 2012 fra <http://www.gnu.org/licenses/quick-guide-gplv3.html>

Software release life cycle - Wikipedia, the free encyclopedia. (u.d.). Hentede 18. juni 2012 fra Wikipedia: http://en.wikipedia.org/wiki/Software_release_life_cycle

Sourceforge. (u.d.). Hentede 11. juni 2012 fra <http://sourceforge.net/about>

Varnish Cache 3.0 beta 1 | Varnish Community. (u.d.). Hentede 28. maj 2012 fra <https://www.varnish-cache.org/releases/varnish-cache-3.0-beta1>

Jeppe Poss (2012). Open source: Kritiske valg

Varnish Software - Online newspapers. (u.d.). Hentede 25. november 2011 fra <https://www.varnish-software.com/references/online-newspapers>

Varnish Software - Social media. (u.d.). Hentede 25. november 2011 fra <https://www.varnish-software.com/references/social-media>

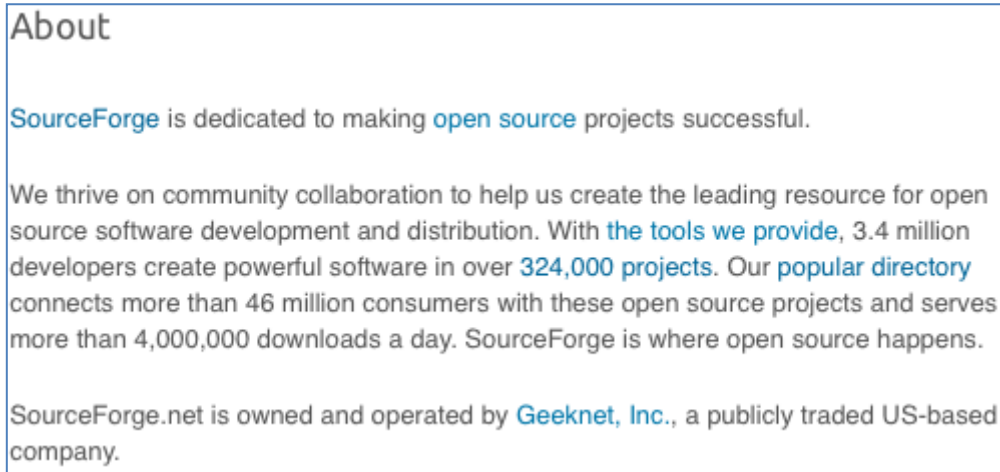
vmods / Varnish Community:. (u.d.). Hentede 31. maj 2012 fra <https://www.varnish-cache.org/vmods>

Welcome to Exherbo Linux. (u.d.). Hentede 10. juni 2012 fra <http://exherbo.org/index.html>

Wikipedia - Varnish software. (u.d.). Hentede 25. november 2011 fra http://en.wikipedia.org/wiki/Varnish_%28software%29

18 Bilag

Bilag 1. **SourceForge.net**



(Sourceforge)

Bilag 2. **Apache.org**

Grundet værkets licens er det ikke muligt at gengive teksten hvori informationerne blev fundet i stedet refereres der kun til værket online, se (Apache Software Foundation).

Bilag 3. **Wikipedia - Varnish cache**

"History

The project was initiated by the online branch of the Norwegian tabloid newspaper Verdens Gang. The architect and lead developer is Danish independent consultant Poul-Henning Kamp (a well known FreeBSD core developer), with management, infrastructure and additional development originally provided by the Norwegian Linux consulting company Linpro. The support, management and development of Varnish was later spun off into a separate company, Varnish Software."(Wikipedia - Varnish software)

Bilag 4. **Varnish Software - Social media**

"The largest global social networking service relies on the support of Varnish Software. Facebook uses Varnish Cache to serve billions of requests every day. As of July 2011, Facebook has more than 750 million active users."(Varnish Software - Social media)

Bilag 5. **Varnish Software - Online newspapers**

"After implementing Varnish Cache in 2006 this

Norwegian online newspaper made a net operating profit of 40 percent, making it an unusually successful online media operation. By 2010 VG sales went upto USD 283 million." (Varnish Software - Online newspapers)

Bilag 6. Bromine README fil

Her er taget et uddrag fra README filen.

"Bromine is an open source QA tool that uses selenium RC as its testing engine.

It provides project management, OS/browser specification, test-case creation as well as user management.

Its lofty aim is to be the open source alternative to commercial tools (such as HP Quality Center).

Bromine also substantially eases the process of creating, maintaining and running Selenium RC tests. The tests are created with the custom Selenium IDE format and uploaded to Bromine. The test can then be run on test machines (we refer to them as nodes) which are configured in Bromine."(Palm, Kløft, & Poss)

Bilag 7. VMods

Her er taget et uddrag websiden.

"Summary of changes from 2.1.5 to 3.0 beta 1

Preliminary streaming support, both on miss and on pass.

Compression and uncompression support, including stitching together compressed ESI fragments.

Plugins (vmods)

Much improved documentation.

Better default values for parameters.

Varnishncsa now has custom log format support.

Varnishlog, varnishncsa and varnishhist can now filter out records that match multiple expressions."

(Varnish Cache 3.0 beta 1 | Varnish Community:)

Bilag 8. VUG1

"Welcome to the World's First VUG meeting

Update: We will start 09:00 London-time on Monday. The meeting is at Canonical's offices in Millbank Tower.

Location: Canonical, Millbank Tower, London, UK

When: September 21st 09:00-> and 22nd, 2009

Who: Varnish administrators and developers". (200909UserGroupMeeting – Varnish:)

Bilag 9. VMods liste

Name	Licence	Status	Varnish version	Commercial support
Authentication	FreeBSD License	Used in production	3.0	
Crashhandler	FreeBSD License	Used in production	3.0	Varnish Software
cURL	FreeBSD License	Used in production	3.0	Varnish Software
dClass	Apache License 2	Used in production	3.0	
DeviceAtlas Mobile Detection	Other	Used in production	3.0	Varnish Software
Digest	FreeBSD License	Used in production	3.0	Varnish Software
example vmod - hello world!	FreeBSD License	Proof of concept	3.0	
Header manipulation	FreeBSD License	Used in production	3.0	Varnish Software
Memcached	FreeBSD License	Used in production	3.0	
null - Binary data in synthetic	FreeBSD License	Used in production	3.0	Varnish Software
redirect	FreeBSD License	In development	3.0	Varnish Software
Redis	FreeBSD License	Proof of concept	3.0	
Secure download	FreeBSD License	Proof of concept	3.0	
Shield	FreeBSD License	Used in production	3.0	Varnish Software
std - the standard VMOD	FreeBSD License	Used in production	3.0	Varnish Software
URL Code	FreeBSD License	Used in production	3.0	Varnish Software
URL Sort	FreeBSD License	Used in production	3.0	Varnish Software
Variable Support	FreeBSD License	Used in production	3.0	Varnish Software

Kilde: (vmods | Varnish Community:)

Bilag 10. Rettigheder i forhold til GPL

I denne artikel angives en række friheder som skal være opfyldt når der tales om gratis software. Grundet artiklens licens er det ikke muligt at gengive ordlyden af disse friheder. I stedet er de omformuleret og brugt som inspiration til de rettigheder brugt i denne afhandling. Hele artiklen kan findes i reference listen under (Smith).

Bilag 11. Exherbo fysiske udtryk: Hjemmeside

På Exherbos hjemmeside findes en række statements i forhold til deres mission og mål samt midler for at opnå nogle af disse mål.

Mission

Exherbos mission bliver udtrykt med disse sætninger:

"Be able to deliver features that other distributions consider too ambitious." citat(Welcome to Exherbo Linux)

"Decentralised development done properly" citat(Welcome to Exherbo Linux)

"We need the freedom to break things when necessary." citat (Welcome to Exherbo Linux)

"Exherbo aims to be a stable and flexible distribution for developers and tinkerers, people who need their distribution to help them be productive. Exherbo actually makes it fun for developers to develop, be it on their own code or the distribution itself. A good set of tools, clear, up-front configuration and excellent support for decentralized development make the system work for instead of against the developer" citat (Dahan). Dette citat er underlagt følgende licens: Copyright 2008, 2009 Jonathan Dahan. This work is licensed under the Creative Commons Attribution Share Alike 3.0 License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-sa/3.0/>. Denne licens er kun gældende for dette citat og det ved ikke gældende for resten af denne kandidatafhandling.

Mål

Projektets mål defineres som:

"Encourage users to take part in the development process and support them with good feedback for their contributions" citat (Welcome to Exherbo Linux).

"No over-centralisation. Only widely used packages are to be in main repositories, and make the tools good enough to deal with lots of small third party repositories for random other apps." citat(Welcome to Exherbo Linux).

Bilag 12. Exherbo repositories

Til beregning af community'ets størrelse er der taget udgangspunkt i følgende repositories.

Exheres

arbor.git
desktop.git
emacs.git
games.git
gnome.git
gstreamer.git
hardware.git
haskell.git
java.git
kde.git
lisp.git
media.git
mono.git
mozilla.git
net.git
office.git
perl.git
php.git
python.git
scientific.git
selinux.git
tcl.git
texlive.git
vim.git
virtualization.git
x11.git
xfce.git

Exheres dev

dev/alip.git
dev/arkanoid.git
dev/berniyh.git
dev/ciaranm.git
dev/compnerd.git
dev/dleverton.git
dev/ferdy.git
dev/ingmar.git
dev/kimrhh.git
dev/marv.git
dev/mixi.git
dev/moben.git
dev/mrothe.git
dev/philantrop.git
dev/pioto.git
dev/pipping.git
dev/rbrown.git
dev/replica.git
dev/sejo.git
dev/sepek.git
dev/spbecker.git
dev/spoonb.git
dev/tanderson.git
dev/woutershep.git
dev/zlin.git

Other

annie.git
dexter.git
docs.git
e4r.git
eclectic.git
exherbo.git
exheres-mode.git
exheres-syntax.git
genesis.git
graveyard.git
infra-scripts.git
infra/hacchi.git
inops.git
rightstart.git
skeleton-filesystem-layout-base-etc.git
summer.git
sydbox.git
unwritten.git
www.git

Kilde: (Exherbo git repositories)

Der er valgt tre repositories fra, da disse ellers ville have skævvredet billedet, da de er oprettet før projektets start. Disse er: git-hooks.git, paludis/paludis-scripts.git og paludis/paludis.git.

Bilag 13. Exherbo Social Engineering

Når BØ taler om social engineering, er det hans måde at styre projektet på. Ledelsesformen bygger på at det skal være nemt for nye at blive en del af community'et. Dette gøres ved at give dem hurtige succesoplevelser. For at give dem den oplevelse, undlader han at rette fejl, som kan betegnes som trivielle. Dette gør at de hurtigt kan bidrage til projektet ved at rette disse trivielle fejl.

Når projektet modtager kontributioner forsøges det så vidt muligt at eksponere dem overfor resten af community'et. Han nævner at dette er særligt vigtigt når der er tale om nytilkommende, da dette bidrager til en bedre oplevelse af at være en del af et community. Selv hvis de nytilkommende har fået betydeligt hjælp til at rette en fejl, er det den nytilkommende der får al æren ved kontributionen. Dette resulterer i at de nye hurtigere bliver optaget i community'et og tager disse værdier med sig og hjælper andre nye der kommer til projektet på samme måde.

Disse tiltag handler i bund og grund om at personer skal stige i status i forhold til den sociale rangstige. Den sociale rangstige kan gøres op i personens aktivitet samt bidrag til projektet.

I Exherbo projektet har BØ forsøgt at gøre det så klart så muligt, hvad han forventer samt hvordan han får folk til at gøre som han anser som værende den rigtige måde. Det at melde dette ud, via dialog, men også ved at blogge omkring dette har været en positiv oplevelse for ham.

Bilag 14. Exherbo Manifest

Et uddrag af manifestet, hvori at Gentoo bliver genstand for kritik.

```
It's not that we think that Gentoo is bad. It's just that we think we can do something that suits our needs better. We've tried, without success, to do this using Gentoo. Unfortunately, Gentoo has serious shortcomings in several areas that stopped this from being a viable long-term approach.

Aspects we find particularly problematic include:

* Portage. The code is too broken and unmaintainable to deliver the changes we need in the timeframe we need. Portage can't deliver even the basic needs of Gentoo; for what we're trying, it's completely out of the question. Unfortunately, Gentoo is too tightly attached to Portage at the expense of the tree.

* Gentoo management.

* QA. Gentoo developers, by and large, don't consider QA to be their problem. Several prolific developers actively flaunt QA, repeatedly break things, refuse to do even simple like keeping a proper changelog and generally screw around with the tree. In Gentoo this is considered acceptable because spending a little time to do things properly wouldn't be fun (never mind the lack of fun for everyone else having to clean up) and because no-one is allowed to tell a volunteer what to do. Similarly, maintainers are free to screw up archs out of carelessness, spite or general stubbornness.

* The users. Not the real users, but the annoying few dozen vocal users, many of whom haven't used Gentoo for years, who go around yelling "Gentoo is about community!" because they don't actually use anything Gentoo provides any more. And, combined with that, the disproportionate attention that is paid to those users' demands as opposed to real users' needs.

* The developers. Not the ones that do things properly. The ones that either do nothing or repeatedly break things.

* Lack of overall design and direction. Gentoo direction is controlled by anyone who can be bothered to spend a few weeks working upon some pet project which is then left as a huge mess for everyone else to handle. There is also a faulty assumption that because it is a volunteer project, developers can't be expected to contribute towards particular project goals.

Ultimately, we believe that Gentoo was a good starting point. However, to provide what we need and what we want, a new generation of tools and development is required.
```

Kilde: (Exherbo Manifest)

Bilag 15. Interview guide

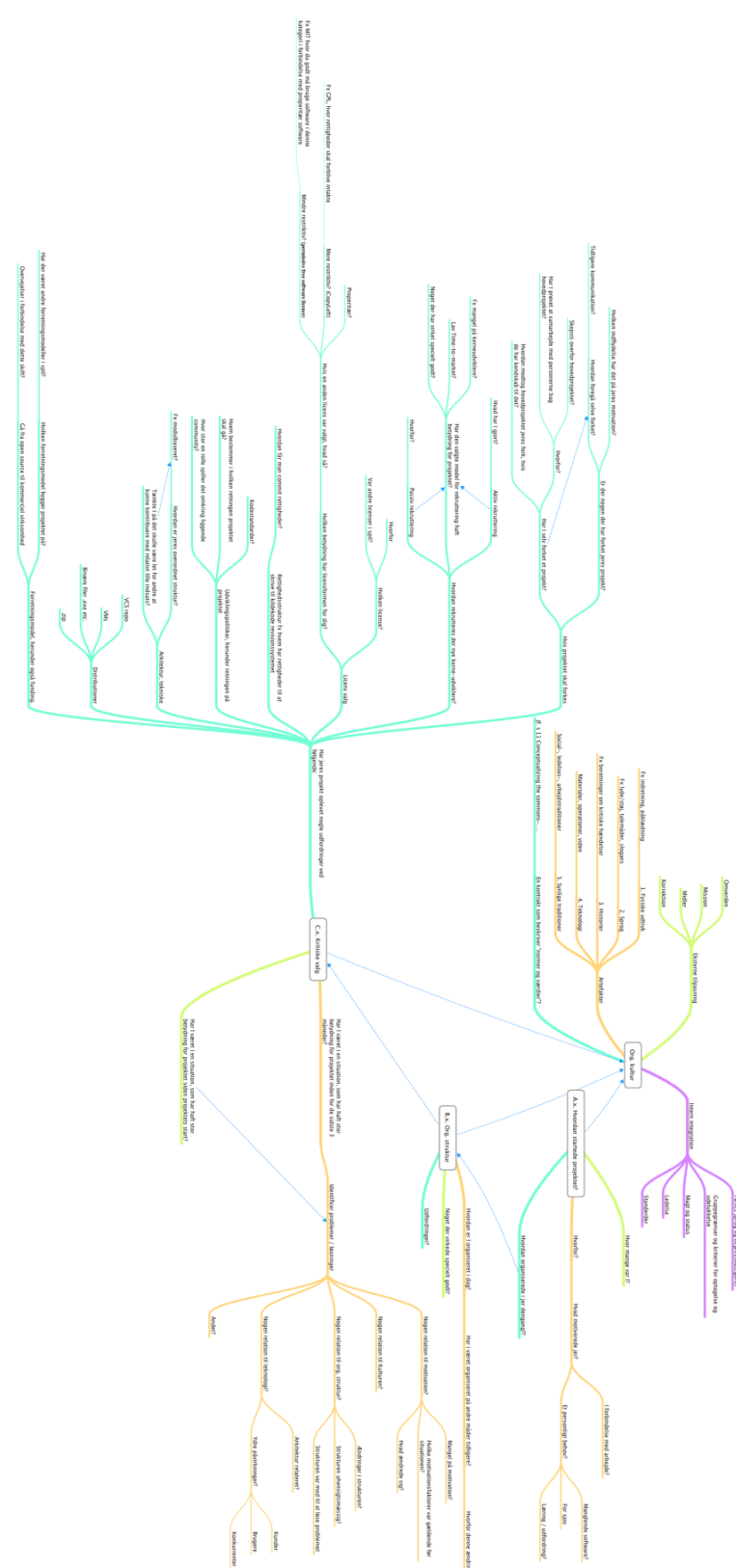
ID	Spørgsmål (Opstart af projekt)	Relation
A.1.	Hvordan startede projektet?	Kultur: Omverden, Mission
A.2.	Hvad motiverede jer?	Motivation Kultur: Midler, magt og status
A.3.	Var det i forbindelse med arbejde eller et personligt behov?	Motivation
A.4.	Hvad udsprang behovet for? (manglende software, for sjov, læring udfordring?)	Motivation Kultur: Omverden, mission
A.5.	Hvor mange var I ved opstarten af projektet?	Kultur: Midler, mission
A.6.	Hvordan organiserede I jer? (Gå til guide omkring org. struktur for at danne et overblik over den initielle organisering)	N/A
A.7.	Hvordan anses community delen.	Kultur: Omverden

ID	Spørgsmål (Organisationsstruktur)	Relation
B.1.	Hvordan er I organiseret? (fx hierarkisk struktur eller basar-lignende hvor alle er lige)	Kultur: Midler, magt og status, ledelse, korrektion
B.2.	Er der nogen som har specifikke arbejdsopgaver? (fx ledelse, support, webmaster)	Midler, ledelse, standarder
B.3.	Nogen udfordringer med denne struktur?	N/A
B.4.	Noget der virkede specielt godt med denne struktur?	N/A
B.5.	Hvordan er I organiseret i dag? (Gennemgå de 4 øverste spørgsmål igen)	N/A
B.6.	Har I været organiseret på andre måder? (Hvis ja, gennemgå de 4 øverste spørgsmål igen)	N/A

ID	Spørgsmål (Kritiske Valg)	Relation
C.1	Har I været i en situation, som har haft stor betydning for projektet inden for de sidste 3 måneder?	Kritisk valg
C.2	Har I været i en situation, som har haft stor betydning	Kritisk valg

	for projektet siden projektets start?	
C.3	Har jeres projekt oplevet nogle udfordringer ved følgende: • Hvis projektet skal forkes • Hvordan rekrutteres der nye kerne-udviklere? • Licens valg • Rettighedsstruktur fx hvem har rettigheder til at skrive til kildekode revisionssystemet • Udviklingspolitikker, herunder retningen på projektet • Arkitektur, tekniske • Distributioner • Forretningsmodel, herunder også funding.	Kritisk valg

Bilag 16. Mind map til interviewguide



Bilag 17. Bromine patch opfordring

John Fa'atua added a comment - 01/Jul/09 11:17 AM	[Permalink] [Hide]
NOTE: These URLs are bogus in the outside world, but are demonstrative of the URI-space suggested. This was found to be far too useful to NOT suggest here...	
Rasmus Berg Palm added a comment - 06/Apr/10 07:40 AM	[Permalink] [Hide]
This will not be implemented unless the community create a patch or it gets alot of votes on usersvoice.	

Et uddrag fra en fejlrapport oprettet i Bromines fejlrapporteringssystem. Her opfordres der til at indgive patches. Kilde: <http://jira.openqa.org/browse/BR-90>

Bilag 18. Release candidate

Release candidate	[edit]
A release candidate (RC) is a beta version with potential to be a final product, which is ready to release unless significant bugs emerge. In this stage of product stabilization, all product features have been designed, coded and tested through one or more beta cycles with no known showstopper-class bug. A release is called code complete when the development team agrees that no entirely new source code will be added to this release. There could still be source code changes to fix defects, changes to documentation and data files, and peripheral code for test cases or utilities.	

Kilde: (Software release life cycle - Wikipedia, the free encyclopedia)

Bilag 19. Varnish Forum

Et uddrag for Varnish Caches forum, hvor man kan se at overskrifterne henviser til problemer.

	Topic / Topic starter
	Sticky: Counter-spam measures by perbu on Wed, 2012-02-29 16:05
	Access to varnish stat by tamilmani on Tue, 2012-06-19 15:50
	problem with sticky session by bitwind on Tue, 2012-06-19 10:18
	Problem with 503 Errors by pdfforge on Thu, 2012-06-14 15:41
	how to modify varnish's user-agent? (varnish itself) by gaingol on Thu, 2012-06-14 09:45
	PURGE on POST by jivkoa on Tue, 2012-06-12 22:16
	Block Based on X-Forwarding-For by bgdanbaba on Fri, 2012-04-13 01:21
	Host included in varnishncsa logging %r by sc2sc on Wed, 2012-06-13 05:44
	Infinite redirection loop by pdfforge on Mon, 2012-05-21 16:35
	LoadBalancing not work by quang on Mon, 2012-05-28 16:21
	How can I fetch and save object after deliver by wadecong on Sun, 2012-06-03 18:16
	Match regular expression using variable by Dippers on Wed, 2012-06-06 13:52

Kilde: (General discussion | Varnish Community)

Bilag 20. Varnish VUG1

We have room for 14 persons in the meeting room (but we can try to get a bigger one)					
(IRC-)nick	Name	Email	Arrival date	Departure date	Notes
mithrandir	Tollef Fog Heen	tfheen@...	2009-09-20	2009-09-22	
Kristian	Kristian Lyngstol	kristian@...	2009-09-20	2009-09-22	
phk	Poul-Henning Kamp	phk@...	2009-09-20/21	2009-09-22	
sky	Artur Bergman	sky@...	2009-09-20	2009-09-22	
alex_	Alex Cacciamani	alex.cacciamani@...			
arpad	Arpad Ray	arpad.ray@...	2009-09-20	2009-09-22	
elro	Laurence Rowe	lrowe@...	2009-09-20/21	2009-09-22	
	Eivind Bengtsson	ebe@...	2009-09-20/21	2009-09-22	
kane	Jos Boumans	jos@...	2009-09-20	2009-09-22	
	Ben Drewel	bdrewel@...	2009-09-20	2009-09-22	
hX8	Marco Walraven	m.walraven@...	2009-09-20	2009-09-22	
anordby	Anders Nordby	anders@...	2009-09-21	2009-09-22	
helgi	Helgi Þormar Þorbjörnsson	helgi@...	2009-09-20	2009-09-22	
mio	Mario Carvalho	mariocar@...	2009-09-20	2009-09-22	

Kilde: (200909UserGroupMeeting – Varnish:)

Bilag 21. Bromine kritiske valg oversigt

Id	Tidspunkt	Version	Kritisk valg	Community	Org.	Implikationer	Kritisk
K1	Medio - ultimo 2007	0-1	Licensvalg (GNU GPL v2)	N/A ¹⁰²	5	Senere i projektet fik det stor betydning, da udviklerne følte de havde ejerskab over softwaren.	5
K2	Medio - ultimo 2007	0-1	Arkitekturen var en hæmsko i forhold til at kunne drage fuld nytte af informations-handlingsforde- len.	N/A	5	Gjorde softwaren mere kompleks end nød- vendig.	1
K3	Medio	0-1	Autonom ledel-	N/A	5	Var med til at styrke	5

¹⁰² N/A relaterer til, at projektet kun var tilgængeligt i Eniro, hvor der ikke var tale om et decideret community i OSS henseende.

	2007		se			ejerskabet over projektet for udviklerne. Fik negative konsekvenser senere i projektet, da det blev forsøgt at rekruttere nye medlemmer.	
K4	Maj 2008	1.9.0.1 beta	Optagelse ved OpenQA	4	4	Første officielle udgivelse. Ved at blive del af et stort community, blev der generet en del opmærksomhed omkring projektet.	5
K5	September 2008	2	Distributioner: Valgte at udvide kanalerne med også at inkludere en self-installer.	+8	4	Gjorde det nemmere for brugerne at komme i gang med softwaren. Flyttede en del af supporten fra installation til brug af softwaren.	1
K6	August 2009	3 beta	Arkitektur: Skrevet hele kodebasen om. Brug af CakePHP.	+12	4	Åbnede arkitekturen op ved at benytte et rammeværk samt understøttelse af plugins.	2 (4)
K7	April 2010	3 RC 1	Ledelsesmæssig brud i forbindelse med stoppet hos Eniro	+47	3	Mangel på ledelse og en person, som tog det overordnede ansvar. Dette var begyndelsen på faldende motivation blandt udviklerne.	1
K8	September 2010	3 RC 2	Distributioner: Suppleret med	+57	3	Gjorde det meget nemmere for bruger-	3

			virtual ap- pliance.			ne, og eliminerede en væsentlig del af sup- portbyrden.	
K9	Novem- ber 2011	3 RC 2	Projektet stop- pede officielt.	+67	3	Det meldtes ud at projektet stoppede og folk opfordredes til at overtage projektet.	5

Bilag 22. Varnish kritiske valg oversigt

Id	Tidspunkt	Version	Kritisk valg	Community	Org.	Implikationer	Kritisk
K1	Primo 2006	0	Licensvalg: FreeBSD	Mindst 3	3	Alle kunne bruge soft- waren uden restriktioner.	2
K2	Primo 2006	0	Kultur: Mål "vi skal være de bedste"	Mindst 3	3	Svært at rekruttere, da der ikke blev gået på kompromis med kvalite- ten.	3
K3	Primo 2006	0	Distributioner: Nemt at instal- lere	Mindst 3	3	Distributionerne var la- vet, så det var nemt at komme i gang med.	3
K4	Primo 2006	0	Kode hygiejne	Mindst 3	3	Gør, at der ikke var stum- per af kode, som ingen vidste, hvad gjorde. Høj test coverage havde po- tentiale for at lette over- gange med nye udviklere.	4
K5	Septem- ber 2009	2.0.4	Brugerinddra- gelse. VUG	+180	6	Anden type brugerdelta- gelse end CBPP foreskrev.	3
K6	Maj 2011	3.0 beta 1	Arkitektur (modularitet) VMods	+365	6	Åbnede op for arkitektu- ren. Havde et stort po- tentiale, da det kunne være med til, at flere satte sig ind i kildekoden, som kunne betyde, at	2 (5)

						flere (frivillige) tilsluttede sig projektet.	
K7	Ultimo 2011	3.0.2	Licensvalg: APJ valgte ikke at bruge Varnish Softwares bog.	+445	6	Varnish bogen, som blev udgivet af Varnish Software, var under en anden licens (Creative Commons), hvilket betød, den ikke måtte bruges kommercielt. Dette havde fået konsekvenser i den forstand, at APJ ikke havde læst den og ikke ønskede at benytte den i forbindelse med den officielle dokumentation.	2
K8	N/A	N/A	Bestyrelse	N/A	N/A		(4)

Bilag 23. Exherbo kritiske valg oversigt

Id	Tidspunkt	Version	Kritisk valg	Community	Kerne-gruppen	Implikationer	Kritisk
K1	Medio ultimo 2007	N/A	Kultur: Bevidst om projektets værdier	+7	+7	Har styrket kulturen i projektet.	5
K2	Medio 2008	N/A	Kultur: Skift til git	+37	29	Skiftet betød en forbedring af projektets belønningssystem.	1
K3	Februar 2009	N/A	Kultur: Mani-fest ¹⁰³	+51	34	Et kulturelt brud med, hvordan omverdenen opfattes	2

¹⁰³For yderligere information henvises til Exherbos git repository "www.git" commit id "13503f312d5585388bac427d10c10bb4fb640599" (Exherbo git repositories)

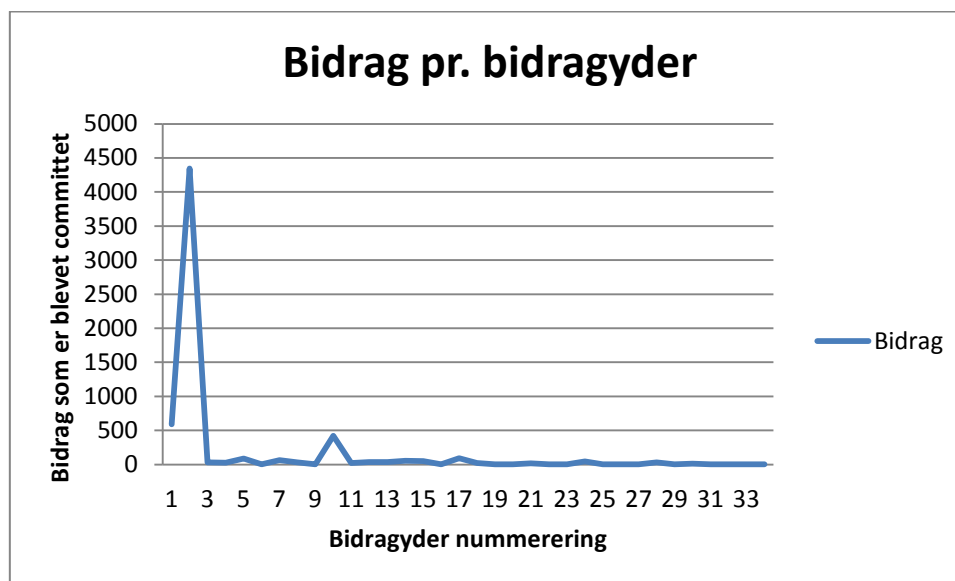
							af medlemmerne.
K4	Oktober 2010	N/A	Kultur: ninger	Forvent-	+138	27	Bedre filtrering af 2 medlemmer.

Bilag 24. Interview

Forkortelse	Navn	Projekt	Position / ansvar
RBP	Rasmus Berg Palm	Bromine	Kerneudvikler
PVK	Peter Visti Kløft	Bromine	Kerneudvikler
CN ¹⁰⁴	Christian Nørlyng	Bromine	Testmanager
PHK	Poul-Henning Kamp	Varnish Cache	Arkitekt / kerneudvikler / gatekeeper
APJ	Andreas Plesner Jacobsen	Varnish Cache	Dokumentation
BØ	Bryan Østergaard	Exherbo	Projektejer / kerneudvikler

Bilag 25. Varnish antal bidrag

Grafen skitserer hvor mange den enkelte udviklere har bidraget med i forhold til commits.



¹⁰⁴ Interview i form af e-mail korrespondance.