

# Deltagande inom open source

*Icke-utvecklarens deltagande i utvecklingsprocessen*

Anders Vennergrund  
2014

Filosofie kandidatexamen  
Systemvetenskap

Luleå tekniska universitet  
Institutionen för system- och rymdteknik

## **Sammanfattning**

Syftet med denna studie har varit att undersöka hur riktlinjer inom traditionella open source-projekt beskriver att användare utan programmeringskunskaper, icke-utvecklare, kan delta i utvecklingsprocessen. Undersökningen har gjorts i form av en fallstudie av projektet Freeplane, ett projekt som har för avsikt att göra alla röster hörda i utvecklingsprocessen samt öka användbarheten hos den mjukvara projektet utvecklar. Den metod som valts är kvalitativ textanalys av projektets wiki.

Studien kommer fram till att projektet har för avsikt att göra icke-utvecklarna delaktiga i utvecklingsprocessen och att öka användbarheten i projektets mjukvara, men att det i projektet finns ett utvecklarperspektiv där utvecklaren ges större utrymme i utvecklingsprocessen. Inom traditionell open source-utveckling har utvecklaren ofta varit central och de mjukvaror som utvecklats påbörjats som lösningar av utvecklarnas egna problem. Studien föreslår därför att icke-utvecklarna inkluderas tidigare i utvecklingsprocessen.

## **Abstract**

The objective of this study was to examine how guidelines in traditional open source projects describe how users without programming skills, non-developer users, can participate in the development process. The survey was conducted as a case study of the project Freeplane, a project that intends to make all voices heard in the development process and increase the usability of the software that the project develops. The method selected was a qualitative text analysis of the project wiki.

This study concludes that the project intends to make non-developer users involved in the development process and increase the usability of the software developed. However, the developer is given more space in the development process. The developer has often been central in traditional open source development since the software development often begin as a solution to the developers' own problems. The study therefore suggest that the non-developer users should be included earlier in the development process.

# Innehållsförteckning

<b>Sammanfattning</b>	<b>2</b>
<b>Abstract</b>	<b>3</b>
<b>Innehållsförteckning</b>	<b>4</b>
<b>1 Inledning</b>	<b>6</b>
1.1 Bakgrund . . . . .	6
1.2 Problemområde . . . . .	7
1.3 Syfte . . . . .	8
1.4 Forskningsfråga . . . . .	8
<b>2 Teori</b>	<b>9</b>
2.1 Open source och fri programvara . . . . .	9
2.1.1 Open source som community . . . . .	10
2.2 Deltagande design . . . . .	10
2.2.1 Icke-utvecklare och utvecklare . . . . .	11
2.2.2 Deltagande . . . . .	12
2.2.3 Direkt och indirekt deltagande . . . . .	13
2.2.4 Roller vid deltagandet . . . . .	14
2.2.5 Kommunikationsverktyg . . . . .	14
2.2.6 Design . . . . .	15
2.3 Deltagande inom open source . . . . .	15
2.3.1 Motivation till att delta . . . . .	16
<b>3 Metod</b>	<b>17</b>
3.1 Forskningsansats . . . . .	17
3.2 Forskningsstrategi . . . . .	17
3.3 Fallstudie . . . . .	17
3.4 Urval . . . . .	18
3.5 Textanalys . . . . .	19
3.6 Validitet och reliabilitet . . . . .	20

3.7	Frågeschema . . . . .	21
<b>4</b>	<b>Resultat</b>	<b>23</b>
4.1	Projektets användargrupp . . . . .	23
4.2	Begrepp i projektet för icke-utvecklare och utvecklare . . . . .	25
4.3	Beskrivning av projektet gentemot dess användargrupper . . . . .	28
4.4	Möjliga sätt för deltagande i projektet . . . . .	34
<b>5</b>	<b>Analys</b>	<b>40</b>
5.1	Utvecklingsprocessen . . . . .	40
5.2	Projektets community . . . . .	41
5.3	Icke-utvecklare och utvecklare . . . . .	41
5.4	Deltagande . . . . .	42
5.4.1	Direkt och indirekt deltagande . . . . .	43
5.4.2	Roller vid deltagandet . . . . .	43
5.5	Möjliga sätt för icke-utvecklare att delta i projektet . . . . .	44
<b>6</b>	<b>Slutsats och diskussion</b>	<b>45</b>
6.1	Metoddiskussion . . . . .	47
6.2	Fortsatt forskning . . . . .	48
<b>7</b>	<b>Referenser</b>	<b>49</b>

# 1 Inledning

I inledningen redogörs för undersökningens bakgrund och problemområde. Detta följs av studiens syfte och forskningsfrågor. Kapitlet avslutas med studiens avgränsningar.

## 1.1 Bakgrund

Open source och fri programvara bygger på att människor deltar i projekt på frivillig basis. Inom denna rörelse finns det dock ett antal olika sätt att se på deltagande. Begreppet fri<sup>1</sup> programvara myntades av Richard Stallman som grundade Free Software Foundation. Datortillverkare kan i hans mening exempelvis bidra med hårdvara och pengar, individer kan bidra genom att donera programvara och arbete. Grundtanken är att alla skall kunna få tillgång till ett fritt, väl fungerande system som användaren får köra, kopiera, distribuera, förändra och förbättra (Stallman, 2002). En förutsättning för att kunna nyttja samtliga friheter i Stallmans synsätt är dock att användaren också har programmeringskunskaper.

I sin essä *Katedralen och basaren*, beskriver Raymond (2001) två typer av utvecklingsmetoder för fri programvara. Katedralen innebär att mjukvaran utvecklas av ett fåtal utvecklare och källkoden ges ut samtidigt som en ny version släpps. I basarmodellen är ofta källkoden hela tiden tillgänglig och utvecklingen sker öppet via internet. Grundtanken är att ju fler som tittar på koden, desto snabbare kan buggar upptäckas och åtgärdas. Raymonds synsätt utgår även detta ifrån ett utvecklar-perspektiv vilket förutsätter programmeringskunskaper, där han också menar att de bästa programmen börjar som lösningar på utvecklarens<sup>2</sup> egna vardagliga problem. En blivande utvecklare måste enligt Raymond därmed lära sig programmera, använda och lära sig en öppen Unix-variant, kunna skriva html-kod, skriva program med öppen källkod och hjälpa till att testa och fixa buggar. Det är också möjligt att bidra genom att publicera nyttig information och underhålla infrastrukturen (Raymond, 2001).

I ovan nämnda synsätt ligger ett stort fokus på projekts kod och utvecklaren, vilket också medför att deltagaren i det traditionella open source-projektet förväntas ha programmeringskunskaper och att denne bidrar med programmeringskod eller åtgärdar buggar. Ofta är också utvecklarna, liksom Raymonds beskrivning av att bra programvaror börjar som lösningar på utvecklarens

---

<sup>1</sup>Begreppet "fri" avser frihet, inte pris (Stallman, 2002)

<sup>2</sup>Raymond benämner utvecklaren "hacker"

problem, också användare av den programvara de utvecklar. Något som kan beskrivas som att ”they scratch their own itch” (Stalder, 2005; Iivari, 2011). I och med att de programvaror som utvecklas inom traditionella open source-projekt används av allt fler användare som inte vill delta i programmeringen, icke-utvecklare, blir det allt viktigare att tillgodose dessa användares behov i design- och utvecklingsprocessen.

## 1.2 Problemområde

Deltagande design inom open source har inte undersökts särskilt mycket till dags dato men open source anses förespråka en viss typ av deltagande design, där en viktig styrka är användardeltagande. De studier som gjorts tidigare fokuserar dock på utvecklaren och utvecklare som använder programvaran (Iivari, 2011). För att delta i open source är det dock viktigt att förstå communityn, dess kultur och de verktyg som användas (Goldman & Gabriel, 2005).

Inom open source är det svårt att särskilja mellan utvecklare och användare, detta då utvecklarna av programvaran ofta förutsätts vara användare av den programvara de utvecklar (Iivari, 2011). Iivari (2011) menar att upp till 90% av användarna i open source-projekt kan vara ”passiva”, vilket innebär att de inte vill delta i programmeringen. I denna studie används begreppet icke-utvecklare<sup>3</sup> för att benämna dessa användare.

Då open source-mjukvaror ofta utvecklas av- och för utvecklare och avancerade användare som också använder mjukvaran, har open source kritiserats för att inte beakta användbarhet och att detta inte setts som något viktigt (Andreasen, Nielsen, Schrøder & Stage, 2006; Iivari, 2011). Även inom open source-communityn finns ett stort fokus på utvecklaren, vilket exempelvis framkommer i Stallmans (2002) och Raymonds (2001) beskrivningar av open source och fri programvara. Andreasen m. fl. (2006) menar dock att utvecklare inom open source är intresserade av användbarhet, men att dessa har en begränsad förståelse för vad det innebär och att det inte har någon hög prioritet i praktiken. I och med att open source-programvaror blivit allt mer välkända och används av fler än utvecklarna själva, bör också icke-utvecklarna ges möjlighet att delta i utvecklingsprocessen och få sina behov tillgodosedda.

Tidigare forskning har tittat på vilka verktyg som gör det möjligt att delta i open source-projekt. Exempelvis fokuserar Iivari (2011) på de verktyg som gör det möjligt att delta då hon undersökt

---

<sup>3</sup>Iivari (2011) använder begreppet Non-Developer Users.

ett traditionellt OS-projekt samt ett där utvecklingen sker av ett kommersiellt företag. För att delta i open source-projekt måste dock deltagaren först få en förståelse för hur denne får och kan delta i projekten. I de flesta projekt finns därför riktlinjer och guider som beskriver hur deltagande i projekten är möjligt samt vilka verktyg som finns tillgängliga för detta deltagande. Denna studie kommer därför att fokusera på hur riktlinjer i traditionella open source-projekt beskriver att icke-utvecklare kan delta i utvecklingsprocessen.

I denna studie undersöks det traditionella open source-projektet Freeplane. Freeplane<sup>4</sup> är en mjukvara för att skapa mindmaps som bygger vidare på projektet FreeMind<sup>5</sup>. Projektet Freeplane startades då FreeMind-projektet bland annat ansågs vara dåligt på att fånga upp de bidrag som communityn bidrog med, vilket exempelvis syns i projektets vision nedan.

The main goal of the project from its very beginning is creating a contribution friendly climate. If somebody like you and me is crazy enough to spend time and efforts for the project doing coding, documenting, testing or supporting the community, it should be welcome. The people should have a feeling of being wanted and welcome (Freeplane, 2011d).

### 1.3 Syfte

Syftet med denna undersökning är att få en förståelse för hur riktlinjer i traditionella open source-projekt beskriver att icke-utvecklare kan delta i utvecklingsprocessen.

### 1.4 Forskningsfråga

För att undersöka hur riktlinjer i traditionella open source-projekt beskriver att icke-utvecklare kan delta i utvecklingsprocessen har jag valt att ställa följande forskningsfråga:

- Hur beskriver riktlinjer inom traditionella open source-projekt att icke-utvecklare kan delta i utvecklingsprocessen?

---

<sup>4</sup><http://freeplane.sourceforge.net>

<sup>5</sup><http://freemind.sourceforge.net>



## 2 Teori

I detta kapitel redogörs för undersökningens teori.

### 2.1 Open source och fri programvara

Kollaborativt och användarlett skapande av innehåll har genom internet-communitys lett till användbara resurser, exempelvis open source-programvara (Bruns, 2012). En grundstomme i open source och fri programvara är att människor deltar i projekt frivilligt och tillsammans skapar innehåll och mjukvaror som är fria att användas av alla. Stallman (2002), som myntade begreppet fri programvara och grundade Free Software Foundation, menar att alla skall kunna få tillgång till fritt (som i frihet), väl fungerande system som användaren får köra, kopiera, distribuera, förändra och förbättra.

Raymond (2001) beskriver två utvecklingsmetoder gällande fri programvara. Dels katedralen, vilket innebär att utvecklingen sker av ett fåtal utvecklare som sedan släpper källkoden i och med att en ny version av mjukvaran släpps. I basarmodellen sker utvecklingen istället öppet över internet och källkoden är hela tiden tillgänglig. Tanken är att, ju fler som tittar på koden desto snabbare är det möjligt att åtgärda de problem och buggar som finns i källkoden. Raymond (2001) menar att de bästa programmen är ett svar på utvecklarens<sup>6</sup> egna problem. För att kunna göra detta menar Raymond (2001) att en blivande utvecklare måste lära sig och använda en öppen Unix-variant, lära sig programmera, skriva program med öppen källkod och hjälpa till med att testa mjukvaran och fixa buggar.

Oreg och Nov (2008) beskriver två typer av open source-initiativ. Dels open source software och open source content. Open source software baseras på att frivilliga eller medlemmar i en organisation tillsammans utvecklar en mjukvara där projektet har en ägare som påbörjar projektet. Denne har också rätt att distribuera nya versioner av mjukvaran. De bidrag som deltagaren ger till projektet kontrolleras av projektets ägare och om koden är tillräckligt bra accepteras denna av projektet. Deltagaren får sedan ära för detta bidraget (Oreg & Nov, 2008). I open source content-projekt, exempelvis Wikipedia, bidrar frivilliga med tid och kunskaper för att skapa en produkt tillgänglig för alla. Detta initiativ involverar dock i normala fall inte en review-process

---

<sup>6</sup>Raymond (2001) benämner utvecklaren ”hacker”

och ofta har projekten inte samma krav på expertis som open source software. Det enda som krävs är grundläggande datorkunskaper (Oreg & Nov, 2008).

### **2.1.1 Open source som community**

Deltagarna skapar mer eller mindre tydliga strukturer, beroende på vilken typ och djup av interaktionen som finns med plattformen. När dessa strukturer antar en mer permanent form benämns communitys. Om strukturerna snarare är obeständiga kan detta istället sägas handla om en crowd (Bruns, 2012). Communitys existerar sedan kring en kärna av aktiva och engagerade deltagare med delade värderingar, praktiker och kunskap som dessa har, vilka sedan utvecklas över tid. Utanför denna kärna med de mest aktiva deltagarna finns flera lager allt mindre engagerade deltagare där de delade värderingarna blir mindre tydliga. Förändringarna av dessa värderingar sker också i snabbare takt i den yttre delen av communityn än hos dess kärna (Bruns, 2012).

Deltagandet i online-communities begränsas enbart av deltagarnas internet-tillgång, tillgänglig tid och relevanta kunskaper. Även om dessa barriärer enligt Bruns (2012) kan ses som icke-triviala, så är de samtidigt lättare att komma över än exempelvis de barriärer som finns inom amatörastronomi, där kraven på material och teknik är större.

## **2.2 Deltagande design**

Deltagande design har sitt ursprung i 1960- och 70-talets sociala och politiska folkrättsrörelser (Simonsen & Robertson, 2013). I västvärlden ville allt fler människor få mer att säga till om när det gällde beslut som påverkade deras egna liv. I de rörelser som växte fram var många människor också beredda att delta i kollektiva handlingar för att uppnå gemensamma mål och intressen (Simonsen & Robertson, 2013). Detta var något som uppmärksammades av designers och forskare som försökte hitta sätt för människor att delta i designen av den mänskligt uppbyggda miljön runt i kring dem (Simonsen & Robertson, 2013).

Utveckling av informationsteknik inom deltagande design har framförallt växt fram i Skandinavien, men även i övriga Europa, sedan 70-talets arbetsplatsdemokrati-rörelse. I och med att datorn började införas på många arbetsplatser fokuserade man på hur bättre verktyg kunde utvecklas för arbetstagarna, så att de arbetsmoment som var väldigt repetitiva kunde automatiseras

och de istället kunde utveckla sina färdigheter (Simonsen & Robertson, 2013).

Enligt den skandiaviska traditionen inom deltagande design är det viktigt att de som skall använda informationstekniken också har en central roll i dess design (Simonsen & Robertson, 2013). Inom deltagande design finns ett fokus på hur i kollaborativa utvecklingsprocesser kan möjliggöra deltagande för de som i framtiden kommer att använda tekniken (Simonsen & Robertson, 2013).

Simonsen och Robertson (2013) definierar deltagande design på följande sätt:

a process of investigating, understanding, reflecting upon, establishing, developing, and supporting mutual learning between multiple participants in collective 'reflection-in-action'. The participants typically undertake the two principal roles of users and designers where the designers strive to learn the realities of the users' situation while the users strive to articulate their desired aims and learn appropriate technological means to obtain them.

### **2.2.1 Icke-utvecklare och utvecklare**

Som nämnts tidigare så är det svårt att skilja mellan användare och utvecklare inom open source-utveckling då alla användare också kan ses som potentiella utvecklare (Iivari, 2011). Iivari (2011) menar dock att 90% av användarna skulle kunna vara passiva, vilket innebär att dessa inte vill delta i programmeringen av mjukvaran som utvecklas. Dessa användare benämns icke-utvecklare.

Utvecklare å andra sidan har programmeringskunskaper. I denna studie avser begreppet utvecklare deltagare som innehar programmeringskunskaper och är utvecklare i projektet, oavsett om utvecklaren är användare av mjukvaran eller inte.

I deltagande design görs en uppdelning mellan deltagare i form av användare och designer, där deltagarna hela tiden försöker lära sig av varandra. Begreppen användare och designer likställs i denna studie med icke-utvecklare respektive utvecklare då de beskriver samma typ av användare.

För icke-utvecklarna i deltagande design handlar deltagandet om att lära sig uttrycka sina mål och vilka tekniska förutsättningar och möjligheter som finns för att göra detta. Utvecklaren å andra sidan försöker förstå icke-utvecklarens behov och situation (Simonsen & Robertson, 2013). Uppdelningen är dock inte helt enkel att göra, då bland annat användningen av tekniken också skiljer sig mellan olika typer av deltagare. Mycket förenklat kan icke-utvecklaren sägas

vara den som kommer att använda sig och interagera med den informationsteknik som designas - medan utvecklaren är, likt en arkitekt, ansvarig för designprojektet (Simonsen & Robertson, 2013).

Simonsen och Robertson (2013) menar att det finns två viktiga aspekter inom deltagande design gällande icke-utvecklare och utvecklare. Dels har deltagande design för avsikt att göra det möjligt för de som kommer att använda en teknik att också få sin röst hörd i utvecklingsprocessen, dels kan inte de som inte är professionella utvecklare definiera vad de vill få ut av utvecklingsprocessen om de inte vet vad som är möjligt att göra. För att användarna skall få sina röster hörda kan man i utvecklingsprocessen använda sig av verktyg för att underlätta detta, exempelvis mock-ups och prototyper (Simonsen & Robertson, 2013). Genom hela utvecklingsprocessen behövs också ett gemensamt lärande och utbyte av kunskap.

### 2.2.2 Deltagande

Inom deltagande design är användarnas delaktighet central i flera steg i designprocessen, exempelvis då man försöker förstå användarnas behov, definiera funktionalitet eller förbättra användbarheten (Hess, Randall, Pipek & Wulf, 2013).

Deltagande inom deltagande design handlar om att icke-utvecklarna inte enbart skall vara informanter som svarar på intervjufrågor, utan att de får en erkänd roll i utvecklingsprocessen där de får utrymme att delta i diskussioner och reflektioner. Exempelvis menar Simonsen och Robertson (2013) att detta kan ske genom att icke-utvecklarna ges möjlighet att stå framför en whiteboard tillsammans med andra kollegor och utvecklare för att beskriva hur arbetet med designen fortskrider ur icke-utvecklarens perspektiv, vilket är något som kräver en god relation deltagare emellan.

Icke-utvecklare kan antingen vara involverade i utvecklingsprocessen redan i ett tidigt stadium då krav samlas in och ställs samman, eller i en senare del av utvecklingsprocessen genom exempelvis reflektioner kring mockups. Helst skall dock deltagandet vara kontinuerligt, men detta är dock svårt att uppnå (Hess m. fl., 2013). För att uppnå en ömsesidig läroprocess mellan utvecklare och icke-utvecklare är ett kontinuerligt deltagande ett krav (Hess m. fl., 2013).

Att inkludera deltagare med domänkunskap i utvecklingsprocessen är viktigt då det innebär förbättrad kvalitet på den produkt som utvecklas (Simonsen & Robertson, 2013). Deltagan-

de förespråkas också inom design av informationsteknik, exempelvis inom människa-dator-interaktion, ofta med hjälp av mock-ups och prototyper. Simonsen och Robertson (2013) menar att det finns också politiska och pragmatiska aspekter kring deltagande inom deltagande design. De pragmatiska handlar om att det är viktigt att icke-utvecklare och utvecklare lär sig tillsammans om vilka möjliga tekniska lösningar som finns och hur dessa kan vara användbara, något som är en central del genom hela utvecklingsprocessen. Utvecklare behöver få förståelse för hur informationstekniken kommer att användas samt icke-utvecklarnas behov och arbetssätt. Icke-utvecklarna å andra sidan behöver kunskap kring vilka valmöjligheter som finns när det gäller de tekniska lösningarna, något som utvecklare kan tillgodose användarna med (Simonsen & Robertson, 2013). De politiska aspekterna handlar om att alla röster måste få höras och att det kan ses som en grundläggande rättighet för icke-utvecklaren att få påverka designen av den informationsteknik denne använder (Simonsen & Robertson, 2013).

### **2.2.3 Direkt och indirekt deltagande**

Deltagande design har ett stort fokus på design där samspelet mellan icke-utvecklare och utvecklare betonas och båda bidrar till designen av den framtida teknologin. Deltagandet kan ske direkt av icke-utvecklaren eller indirekt via representanter<sup>7</sup>(Iivari, 2011). Dessa representanter kan antingen sägas representera icke-utvecklarna i utvecklingsprocessen eller förenkla samspelet mellan icke-utvecklare och utvecklare. Dels kan representanterna inneha formella positioner i projektet, dels informella. De förväntas möjliggöra deltagande, se till att allas expertis uppskattas och att alla känner sig bekväma med att delta (Iivari, 2011). I utvecklingsprocessen kan representanterna få en mer aktiv roll där de själva aktivt deltar och har makt att också ta beslut i denna. De kan också ges en mer begränsad roll där de exempelvis enbart ger kommentarer på fördefinierade designförslag. Det är däremot inte alltid självklart att varken icke-utvecklarna eller dess representanter har någon påverkan på själva lösningen, även om det förespråkas att dessa får en aktiv roll i processen (Iivari, 2011). Exempelvis HCI specialister skulle kunna hjälpa till med användarstudier och pappersprototyper i ett tidigt skede av utvecklingen, dessa är dock normalt inte med i utvecklingen av open source. Om dessa specialister inte får möjlighet att påverka beslut är det inte säkert att deras förslag påverkar lösningen (Iivari, 2011).

---

<sup>7</sup>vilka betecknas researcher-designers inom deltagande design

## 2.2.4 Roller vid deltagandet

Den distribuerade deltagande design som sker inom open source handlar om att förstå icke-utvecklarnas nuvarande arbetssätt, förändra detta tillsammans med icke-utvecklarna och samla in feedback från dessa för att skapa en lösning. Målet är att förbättra funktionaliteten eller användbarheten när det gäller lösningen och att göra det möjligt för kunniga icke-utvecklare att få en ökad makt i utvecklingsprocessen, där vikten läggs på deltagande i beslutstagandet (Iivari, 2011). I denna process kan både icke-utvecklarna och dess representanter inneha ett antal olika roller, där de dels kan bli observerade och lämna information, kommentera på designförslag, ha en roll som co-designers, producera lösningen tillsammans med utvecklare och/eller ha makt att ta beslut kring lösningen (Iivari, 2011). Representanterna kan också vara delaktiga i designlösningen och inneha liknande roller som icke-utvecklarna. De är också ansvariga för att förstå icke-utvecklarna nuvarande praktiker, designa nya praktiker och teknologier samt samla in feedback (Iivari, 2011).

## 2.2.5 Kommunikationsverktyg

I och med framväxten av nya mer sociala teknologier som exempelvis wikis och bloggar, möjliggörs nya former av deltagande både på distans och online (Hess m. fl., 2013).

Inom deltagande design har det undersöks olika metoder och verktyg för att mer aktivt inkludera icke-utvecklare i utvecklingsprocessen. Undersökningarna har visat på ett antal problemområden, exempelvis till vilken grad det är möjligt att inkludera icke-utvecklare i hela utvecklingscykeln, problem med organisationskomplexitet, heterogena uppgifter, balans mellan olika intressenters intressen och ansvarsområden samt problem med att få fram kunskap (Hess m. fl., 2013).

Det kan dock vara svårt för icke-utvecklare att skicka in buggrapporter eller fixa buggar. De kommunikationsverktyg som används kan också verka avskräckande för icke-utvecklare och vara för svåra att använda (Iivari, 2011). När det gäller just användbarhetsproblem kan dessa vara särskilt svåra att kommunicera eftersom denna typ av problem är svåra att kommunicera textuellt (Iivari, 2011). Det är också viktigt att informera både icke-utvecklare och representanter vilken typ av data och vilket format denna skall ha för att den skall vara användbar vid designen (Iivari, 2011). Att uppmuntra icke-utvecklare att spontant och frivilligt bidra med feed-

back och annan information är även detta viktigt, vilket förutsätter att ett forum sätts upp som det finns en aktiv community kring. Enligt Iivari (2011) kommer dock aldrig den stora massan icke-utvecklare ha samma förutsättningar att delta i diskussionen kring teknisk utveckling som de tekniskt kunniga, vilka alltid kommer att ha auktoritet att skapa representationer av de mindre tekniskt kunniga.

### **2.2.6 Design**

Design inom deltagande design innefattar både själva slutprodukten, artefakten, och den process som gör det möjligt att delta i utvecklingsprocessen (Simonsen & Robertson, 2013). Således är en av drivkrafterna inom deltagande design att icke-utvecklare tillsammans med utvecklare skapar och utvecklar idéer och visioner. Inom deltagande design är det därför viktigt hur man designar (Simonsen & Robertson, 2013).

## **2.3 Deltagande inom open source**

Användbarheten inom open source har dock ofta varit bristfällig för icke-utvecklare, eftersom utvecklarna tidigare framförallt utvecklat mjukvaror utifrån sina egna behov. Användbarheten hos mjukvaran har därför sällan stått i fokus (Iivari, 2011). I och med att användarna av mjukvarorna inte längre nödvändigtvis är desamma som utvecklarna, menar Iivari (2011) att det är viktigt att icke-utvecklarnas behov också tillgodoses. Deltagande design har betonats som en styrka hos just open source-utveckling, men de studier som gjorts fokuserar dock på utvecklare som också är användare av den mjukvara de utvecklar (Iivari, 2011).

Ett antal metoder och verktyg har tagits fram för att stödja deltagande design inom HCI och deltagande design. Dessa förutsätter dock att deltagarna finns på samma geografiska plats, vilket normalt sätt inte gäller inom open source-utveckling (Iivari, 2011). Open source är istället en form av distribuerad utveckling, där utvecklingen sker på distans med utvecklare som finns utspridda på många olika geografiska platser, vilket ställer andra krav på deltagande. Deltagandet blir därmed beroende av internet för att deltagarna skall kunna kommunicera och samarbeta, exempelvis genom e-postlistor, diskussionsforum, buggrapporteringsverktyg och versionshanteringssystem (Iivari, 2011). Detta medför att utvecklingen kan ske distribuerat både geografiskt, organisationsmässigt och i tiden, där deltagare över hela världen kan delta i projekt utan

kontrakt och organisationstillhörighet (Iivari, 2011). Verktøygen gör det också möjligt för utvecklare att ge support och samla in feedback (Iivari, 2011). Bloggar har föreslagits inom deltagande design för att snabbt och publikt kunna diskutera designfrågor oavsett geografisk plats, tillsammans med exempelvis mockups och textbeskrivningar, vilket enligt Iivari (2011) snarare erbjuder distribuerat designarbete snarare än användardeltagande.

### **2.3.1 Motivation till att delta**

En uppdelning av vad som motiverar deltagare att delta är att dela upp dessa i inre och yttre motivationer. Yttre motivationer fokuserar på att nyttan med att delta är större än den uppoffring som krävs för att delta. Exempelvis kan yttre motivationer handla om att få förbättrade programmeringskunskaper, att skapa programmeringskod som inte är tillgänglig men som behövs eller att förbättra sin professionella status (Oreg & Nov, 2008). Inre motivationer fokuserar istället på en inre tillfredsställelse hos deltagaren. Exempel på inre motivationer är att deltagaren deltar för att få intellektuell stimulans, att denne deltar för nöjes skull eller att deltagaren har en plikt känsla gentemot att delta. Det kan också handla om osjälviska motiv, altruism, där deltagaren sätter andras intressen framför sina egna (Oreg & Nov, 2008).

Inom mjukvaru-initiativ har de flesta deltagare också mjukvarurelaterade arbeten. Inom content-initiativ är många deltagare däremot inte professionella skapare av innehåll (Oreg & Nov, 2008). Genom att delta i open source-projekt kan deltagare tydliggöra sina kunskaper, sin identitet och sin status. Altruism anses vara en motivation både inom mjukvaru- och innehållskontext som är högt värderat inom communityn. Att briljera med sina förmågor är det däremot inte (Oreg & Nov, 2008).



## 3 Metod

I detta kapitel beskrivs studiens tillvägagångssätt.

### 3.1 Forskningsansats

Denna studie har en deduktiv forskningsansats, vilket innebär att forskaren drar slutsatser om enskilda företeelser utifrån befintliga teorier (Patel & Davidson, 2011). Ett deduktivt angreppssätt anses också kunna stärka forskningens objektivitet då denna blir mindre färgad av forskarens egna uppfattningar (Patel & Davidson, 2011).

### 3.2 Forskningsstrategi

Kvantitativa studier avser de studier som görs genom mätning vid datainsamling som sedan analyseras med statistiska bearbetningar (Patel & Davidson, 2011). I kvalitativa studier fokuserar istället insamlingen av data på ”mjuka” data. Det handlar exempelvis om tolkande analyser och kvalitativa intervjuer (Patel & Davidson, 2011).

Jag har valt att göra en kvalitativ studie då studiens syfte är att få en förståelse för hur riktlinjer i traditionella open source-projekt beskriver att icke-utvecklare kan delta i utvecklingsprocessen.

### 3.3 Fallstudie

En fallstudie innebär att undersökningen görs på en mindre avgränsad grupp, vilken kan bestå av en organisation, situation, en individ eller flera individer (Patel & Davidson, 2011). I denna studie har jag gjort en fallstudie där fallet avser det traditionella open source-projektet Freeplane och dess community.

FreePlane<sup>8</sup> är ett traditionellt open source-projekt som utvecklar en fri programvara för mind-mapping och knowledge management (Freeplane, 2014f). Mjukvaran är släppt under licensen GPL och finns fritt tillgänglig för nerladdning på portalen Sourceforge.net<sup>9</sup>. SourceForge är en resurs för open source-projekt som erbjuder verktyg för utveckling av open source-programvara

---

<sup>8</sup><http://freeplane.sourceforge.net>

<sup>9</sup><http://sourceforge.net>

och möjligheten att publicera, ladda ner och ge omdöme för denna. SourceForge beskriver sig som den största plattformen för Open Source-utveckling på internet, med mer än 430 000 projekt och 4,8 miljoner nedladdningar per dag (Sourceforge, 2014).

Patel och Davidson (2011) menar att fallstudier ofta kan komma till användning vid studier av förändringar och processer. Denna studie avser att undersöka hur riktlinjer i projektet Freeplane beskriver att icke-utvecklare kan delta i Freeplanes utvecklingsprocess.

### 3.4 Urval

Studiens population är traditionella open source-projekt. Med population avses en grupp av fenomen som undersökningen vill uttala sig om, exempelvis personer, händelser, handlingar och institutioner (Esaiasson, Gilljam, Oscarsson & Wängnerud, 2007).

Då studiens syfte är att få en förståelse för hur riktlinjer inom traditionella open source-projekt beskriver att icke-utvecklare kan delta i utvecklingsprocessen, har jag valt att göra ett strategiskt urval. I detta urval har jag valt ut ett kritiskt fall som troligen uppvisar gynnsamma förhållanden för att vilja få icke-utvecklare att delta i utvecklingsprocessen.

Det traditionella open source-projektet Freeplane avser att skapa en miljö som uppmuntrar deltagande, vilket exempelvis syns i projektets vision nedan. Projektet startades då det ursprungliga projektet FreeMind ansågs vara dåligt på att fånga upp bidrag från communityn.

The main goal of the project from its very beginning is creating a contribution friendly climate. If somebody like you and me is crazy enough to spend time and efforts for the project doing coding, documenting, testing or supporting the community, it should be welcome. The people should have a feeling of being wanted and welcome.

As a project member anyone has the freedom to develop his own ideas and document them, to implement them or have a chance to inspire and convince the community and the coders to implement the ideas. Also if somebody wants to contribute in a way not foreseen in a concept, the contribution is allowed to happen and that the concept should reflect new ideas (Freeplane, 2011d).

I projektet Freeplane har jag valt att analysera projektets samtliga wiki-sidor, utom de som skrivits på annat språk än engelska, varit föråldrade eller saknat text. Jag har valt att titta på wikin då denna innehåller beskrivningar och riktlinjer för hur deltagande kan ske i projektet, samt att denna också fungerar som en hemsida för Freeplane. Därmed kan den också vara den första kontakt som icke-utvecklaren får med projektet.

### 3.5 Textanalys

Kvalitativ textanalys innebär att man försöker få fram en texts viktigaste innehåll. För att göra detta sker in noggrann läsning av texten med fokus både på textens kontext, helhet och dess delar (Esaiasson m. fl., 2007). En anledning till att använda den kvalitativa textanalysen framför kvantitativ innehållsanalys är att vissa delar av texten ses som viktigare än andra (Esaiasson m. fl., 2007). Kvalitativ textanalys kan också fånga upp innehåll i texten som finns dolt under ytan (Esaiasson m. fl., 2007).

Jag har valt att göra en kvalitativ textanalys då studiens syfte är att få en förståelse för hur riktlinjer i traditionella open source-projekt beskriver att deltagande kan ske. För att titta på deltagande inom open source är det exempelvis också möjligt att använda sig av intervjuer och observationer, vilket skulle kunna svara på frågor som exempelvis varför icke-utvecklare väljer att delta i traditionella open source-projekt och på vilket sätt.

För att finna lösningen på det forskningsproblem som ställts börjar textanalysen med att ett antal mer precisa frågor tas fram som skall ställas till texten. Esaiasson m. fl. (2007) menar att svaren till dessa frågor också blir lösningen på forskningsproblemet. Dessa frågor måste därför vara goda empiriska indikationer för det undersökta fenomenet (Esaiasson m. fl., 2007).

För att svara på forskningsfrågan har jag tagit fram ett antal frågor som avses svara på denna. Dessa frågor redovisas och motiveras i ett frågeschema, se punkt 3.7. För kunna ställa relevanta frågor har först jag varit ute och tittat på projektets wiki.

Efter att frågorna formulerats måste man bestämma sig för hur man skall förhålla sig till de svar man får ifrån texterna. Dels kan man bestämma kategorier på förhand eller istället anta ett öppet förhållningssätt. Då bestäms svaren istället beroende på vad som återfinns i det undersökta materialet (Esaiasson m. fl., 2007).

Jag har valt att ha ett öppet förhållningssätt vid textanalysen. Efter att jag besvarat första frågan i min textanalys, "Vilka beskriver projektet som sin användargrupp?", har jag valt att använda två breda kategorier (icke-utvecklare och utvecklare) för de resterande frågorna.

Texterna produceras alltid av någon människa och i textanalysen måste man bestämma sig för vilka aktörers texter som skall analyseras. Dels kan studien vara aktörscentrerad, där det är viktigt vem som säger vad, eller idécentral. Vid idécentrala studier är istället idéerna viktigt snarare än vem som säger vad (Esaiasson m. fl., 2007).

I analysen av texterna har jag valt ett idécentralt angreppssätt snarare än att fokusera på specifika aktörer, detta då studien syftar till att förstå hur det i projektet och dess community beskrivs att deltagande kan ske, snarare än hur vissa enskilda aktörer menar att deltagande kan ske.

När man bestämmer sig för vilket material som man skall leta i kan man välja att göra ett brett eller snävt urval. Esaiasson m. fl. (2007) menar att det optimala är att analysera allt relevant material, men menar samtidigt att man måste ta praktiska begränsningar i beaktning. Det är dock viktigt att tänka igenom hur de val av material som görs påverkar sannolikheten för olika utfall (Esaiasson m. fl., 2007).

Jag har valt att analysera projektets samtliga wiki-sidor, utom de som markerats som angetts som föråldrade (1 st), saknat text (1 st) samt sidor skrivna på annat språk än engelska (6 st). Wikin fungerar, förutom som ren wiki, även som webbsida för projektet vilket kan innebära att denna kan fungera som en första kontakt med projektet som icke-utvecklaren får. Wikin innehåller också riktlinjer för hur deltagande kan ske. Därmed kan Wikin sägas svara på hur riktlinjer i projektet beskriver att deltagande kan ske. Däremot kan en undersökning av projektets wiki inte beskriva hur deltagandet sedan sker i praktiken.

En text är alltid ett resultat av att en eller flera personer vill förmedla någonting till andra personer. I och med att texten refereras och läses får den också konsekvenser för vad människor tänker och vad de gör (Bergström & Boréus, 2012). Text speglar också de föreställningar som finns i de miljöer där texten skrivs, både medvetna och omedvetna föreställningar. Texten återspeglar och reproducerar således exempelvis makt och kan användas för att komma åt relationer mellan individer och grupper utanför själva texten (Bergström & Boréus, 2012).

Projektets wiki torde således avspejla ett antal föreställningar som finns hos projektets community, där det också är möjligt att relationer mellan exempelvis icke-utvecklare och utvecklare avspeglas i texten.

### **3.6 Validitet och reliabilitet**

En mätmetods validitet kan sägas handla om den mäter det den är avsedd att mäta. I en kvalitativ undersökning som inte använder sig av räknande eller mätande handlar validitet istället om den undersökning som genomförs för att besvara en bestämd fråga verkligen besvarar denna (Bergström & Boréus, 2012).

Genom att jag konstruerat ett frågeschema med ett antal frågor, har jag avsett att kunna besvara forskningsfrågan med dessa.

Bergström och Boréus (2012) menar att forskaren inte bara är en iakttagande utomstående person, utan att denne också är med och konstruerar studieobjektet. Om forskaren ökar förståelse om sin egen förförståelse ökar enligt Bergström validiteten i studierna.

Till min förförståelse av open source-projekt hör att jag använt open source-mjukvaror inriktade både mot icke-utvecklare och utvecklare under många år, då jag har en bakgrund lik utvecklare. Jag har också läst mycket om open source både av aktörer inom och utanför open source-communityn.

Reabilitet handlar om att vara noggrann i de mätningar och uträkningar som görs. För en kvalitativ studie å andra sidan handlar detta istället om att vara noggrann i alla delar av undersökningen och att minska antalet felkällor som mycket som möjligt. I exempelvis en textanalysen måste textläsningen vara tillräckligt noggrann för studiens syfte (Bergström & Boréus, 2012).

Jag har gjort en systematisk läsning och tolkning av texterna genom att jag gått igenom fråga efter fråga. För varje fråga har jag sedan tittat på alla sidor en efter en, vilka jag först samlat in i form av pdf-filer och skärmdumpar.

### 3.7 Frågeschema

För att svara på forskningsfrågan *”Hur beskriver riktlinjer inom traditionella open source-projekt att icke-utvecklare kan delta i utvecklingsprocessen?”* har jag valt att ställa följande frågor till texten:

- Vilka beskriver projektet som sin användargrupp?

För att förstå hur icke-utvecklare kan delta i den utvecklingsprocess som sker i projektet, måste vi först få en förståelse för vilka projektet riktar sig till. Om projektet riktar sig till icke-utvecklare snarare än utvecklare, har det troligvis andra förutsättningar för att få icke-utvecklare delaktiga, än om det snarare inriktat sig mer mot utvecklare.

- Vilka begrepp används i projektet för icke-utvecklare och utvecklare?

För att kunna svara på hur icke-utvecklare kan delta i utvecklingsprocessen, måste vi veta vilka begrepp i projektet som används för icke-utvecklare och utvecklare. Genom att veta

vilka begrepp projektet använder, kan vi förstå om texterna i wikin riktar sig mot icke-utvecklare eller utvecklare. Detta är nödvändigt att veta för att exempelvis kunna svara på hur projektet beskrivs gentemot icke-utvecklare och hur dessa kan delta i projektet.

- Hur beskrivs projektet gentemot dess användargrupp?

Innan vi ställer oss frågan ”hur kan deltagande ske i projektet”, måste vi få en förståelse för hur projektet beskrivs gentemot de användargrupper projektet riktar sig till. Exempelvis skulle projektet kunna beskrivas på olika sätt gentemot icke-utvecklare och utvecklare, eller enbart beskrivas ur ett visst perspektiv. Svaret på denna fråga kan sedan användas för att få en djupare förståelse för:

- På vilket sätt beskrivs det att deltagande kan ske i projektet?

För att kunna svara på forskningsfrågan, hur riktlinjerna beskriver att icke-utvecklarna kan delta i utvecklingsprocessen, måste vi få ställa frågan hur projektet beskriver att deltagande kan ske. Denna fråga skulle exempelvis kunna ge svar på vilka roller icke-utvecklare och utvecklare kan ha i projektet, på vilket sätt de kan delta i dessa roller samt exempelvis vilka verktyg som finns för deltagande.

## 4 Resultat

I detta kapitel redovisas studiens resultat.

### 4.1 Projektets användargrupp

Freeplane kan sägas beskriva sin användargrupp som både utvecklare och icke-utvecklare. När det gäller mjukvaran skall denna vara användarvänlig, snabb och tillgänglig både i hemmet, skolan och på arbetet för att hantera information och kunskap. Funktionaliteten i programvaran Freeplane är inriktad mot icke-utvecklare, men det finns också stöd för att skapa egna tillägg och anpassningar av programvaran, vilket är inriktat både mot icke-utvecklare och utvecklare. En del tillägg kräver dock att script skrivs, vilket kan sägas vara mer inriktat mot utvecklare. I projektet menar man att man uppmuntrar deltagande, där var och en sägs kunna utveckla sina egna idéer. Stor vikt läggs dock vid källkodens omstrukturering och att underlätta deltagande för utvecklare, även om icke-utvecklare också får delta.

Freeplane är en mjukvara som kan användas för att dela information, få saker gjort och att tänka kring saker - exempelvis i form av mindmaps.

Freeplane is a free and open source software application that supports thinking, sharing information and getting things done at work, in school and at home (Freeplane, 2014f).

Projektet inriktar sig främst mot icke-utvecklare, även om man menar att Freeplane som projekt erbjuder ett gott stöd även för utvecklare, i form av att det är lätt att utveckla egna add-ons. Man menar också att studerande kan lära sig programmera i en naturlig och stimulerande miljö (Freeplane, 2014f).

Although Freeplane functionality is intended for non-programmers, the Freeplane project offers potential programmers a versatile scripting language. Programmers may easily develop powerful extensions and students are stimulated to learn programming in a natural and stimulating environment. Freeplane also has a powerful mechanism to install packages of scripts, icons, images, language dictionaries for spell checking and other preferences including a different menu structure; and which can be used by non-programmers (Freeplane, 2014f).

Anledningen till att man skall välja just Freeplane framför andra mindmap-programvaror menar man är att projektets programvara Freeplane skall vara så lätt att använda och så snabb att använda som möjligt (Freeplane, 2014f). Programvaran beskrivs också som att denna skall va-

ra lika naturlig och snabb att arbeta i som en text-editor, samtidigt som innehåll skall kunna struktureras enkelt som i ett diagram (Freeplane, 2014f).

Freeplane aims for maximum ease and speed of use. Occupying the middle ground between an editor and a diagramming tool, Freeplane allows the user to add content as quickly and naturally as they would in a text editor, yet producing structured content that can be manipulated as easily as a diagram (Freeplane, 2014f).

Man menar också att man har ett enkelt verktyg för att installera paket med script, ordlistor för olika språk, bilder och ikoner som kan användas av icke-utvecklare (Freeplane, 2014f).

Freeplane also has a powerful mechanism to install packages of scripts, icons, images, language dictionaries for spell checking and other preferences including a different menu structure; and which can be used by non-programmers (Freeplane, 2014f).

Bland de paket som kan laddas ner finns bland annat ett Issue-Based Information System, GTD - Getting Things Done, Study Planner designad för elever och studenter för att planera sina skolaktiviteter och verktyg för att arbeta flera tillsammans på samma mindmap. Det finns också några paket för utvecklare, exempelvis Add-on Developer Tools som kan användas vid utvecklingen av nya paket (Freeplane, 2014b). Man menar dock att alla kan utveckla sina egna paket och tillägg:

Anybody can write an add-on! (see the add-on developer documentation) (Freeplane, 2014b)

Många av de tillägg som finns kräver dock inga programmeringskunskaper enligt projektet, för vissa kan man dock behöva kunna skriva script: (Freeplane, 2014a).

There are several add-ons that don't need any programming know-how, but depending on the purpose of your add-on you might have to write scripts (Freeplane, 2014a).

Projektet har också ett stort fokus på deltagande där man beskriver projektet i form av att det har en aktiv community. Målet är att skapa en mycket bra programvara för att hantera kunskap genom att ge support till varandra inom communityn (Freeplane, 2014f)

Freeplane is a non-commercial project. It lives by our active and vibrant community, where we try to support each other in order to develop and provide an outstanding knowledge management application. Please, feel free to discuss Freeplane and its features with other interested Freeplane users (Freeplane, 2014f).

Att projektet beskriver sin användargrupp som både icke-utvecklare och utvecklare kan exempelvis ses i projektets mål där ökad användbarhet nämns som första punkten, tätt följt av målet



att skapa ett stort och effektivt team med utvecklare. Man menar också att de som kontinuerligt bidrar i projektet skall ha en större bestämmanderätt när det gäller beslut kring projektet samt att man skall ha en vänligt inställd community där alla röster hörs.

Better usability Big and effective developer Team. People like you and me systematically contributing to the project should get more influence on the project decisions. A large, friendly community in which your voice will be heard (Freeplane, 2011a).

För att göra det enklare för nya utvecklare att delta i projektet har också programvarans källkod omstrukturerats, vilket skall utgöra en grund för långsiktig utveckling av programvaran.

Freeplane's code has been refactored considerably to create a highly modular, clear and logical design, to ease entrance into development and to provide a solid foundation for long term development (Freeplane, 2011c).

Även detta kommer fram i projektets vision, där det tydliggörs att projektet avser att underlätta skapandet av tillägg och göra det möjligt för både utvecklare och icke-utvecklare att anpassa programvaran utefter behov, bland annat genom tydliga och väldokumenterade interface:

We strive to develop FreePlane as a lean framework for mind mapping, with clear and well-documented interfaces, allowing for easy extension and customization by programmers, scripters and other non-developers, federating them in a lively community supporting our project vision (Freeplane, 2011d).

Som ovan nämnt kan Freeplane sägas beskriva både utvecklare och icke-utvecklare som sin användargrupp. Mjukvaran som utvecklas kan sägas inriktad mot icke-utvecklare i huvudsak, men att det också erbjuds möjlighet att göra anpassningar och tillägg till mjukvaran, vilket även riktar sig till utvecklare.

## **4.2 Begrepp i projektet för icke-utvecklare och utvecklare**

För utvecklare används ett flertal liknande begrepp för att beskriva utvecklare, till skillnad från begrepp relaterade till icke-utvecklare som snarare kan delas upp i kategorier beroende på vilket sätt icke-utvecklaren deltar.

Tabell 1: Utvecklare Freeplane 1.3 (2012 - 2014)

Name	Role	Contact
Dimitry Polivaev	Founder, Lead Developer	<a href="https://sourceforge.net/users/dpolivaev/">https://sourceforge.net/users/dpolivaev/</a>
Volker Boerchers	Developer	<a href="https://sourceforge.net/users/boercher/">https://sourceforge.net/users/boercher/</a>
Felix Natter	Developer	<a href="https://sourceforge.net/users/fnatter/">https://sourceforge.net/users/fnatter/</a>
Blair Archibald	Developer	<a href="https://sourceforge.net/users/blairarchibald/">https://sourceforge.net/users/blairarchibald/</a>
Hartmut Goebel	Developer	<a href="https://sourceforge.net/users/htgoebel/">https://sourceforge.net/users/htgoebel/</a>
Robert ...	Graphic designer	...

(Freeplane, 2014c)

### Begrepp för utvecklare

I projektet använder man sig av ett antal likartade begrepp för att beskriva utvecklare, där de flesta kan sägas åsyfta samma sak. Det huvudsakliga begreppet som används i projektet för utvecklare, Developer, kan sägas avse en person som kan programmera och delta i allt. Det finns dock ett antal andra begrepp i projektet som i princip avser samma typ av deltagare, exempelvis coder, core-developer, designer, founder, lead developer och programmer. Skillnaden kan sägas vara vilken vikt man lägger vid programmeringen och hur nära utvecklaren befinner sig kärnan av utvecklare i projektet. Core-developer, founder och lead developer kan sägas utgöra kärnan av utvecklare i projektet (se tabell 1), medan coder och designer snarare kan ses som vilken typ av deltagande som sker.

Join the Freeplane team not only as user but maybe as a

- developer (coder and designer)
- tester (writer of tutorials, screencasts and help texts)
- translator (various languages)
- administrator (forum moderator) (Freeplane, 2014f).

### Begrepp för icke-utvecklare

Begrepp relaterade till icke-utvecklare i projektet kan delas upp i ett par kategorier efter vilken form av deltagande som sker. Exempelvis syftar begreppet tester på en deltagare som bland annat skriver guider och hjälptexter eller provar nya releaser av mjukvaran, medan en administrator antingen är forum moderator eller administratör i projektets wiki. Gällande översättningar används dock begreppet translator konsekvent.

Join the Freeplane team not only as user but maybe as a

- developer (coder and designer)
- tester (writer of tutorials, screencasts and help texts)
- translator (various languages)
- administrator (forum moderator) (Freeplane, 2014f).

För icke-utvecklare används också begreppet users. Framförallt används detta då man beskriver mjukvarans funktionalitet och visar exempel på hur den kan användas. Begreppet innefattar också olika nivåer för användare: nybörjare, avancerade och professionella användare. Det framkommer dock aldrig i wikin vad respektive nivå innebär i praktiken.

This Freeplane Tutorial Extensions describes a Tutorial with basic terms and how to use the basic Freeplane functions. In addition it contains examples of use cases and concrete instructions how to make these cases. The Tutorial is divided into parts for the Beginner, Advanced user and Professional user. The examples below are labelled in a similar way helping to indicate the level of difficulty if you want to make a map yourself. It is no problem if the beginner wants to navigate a map for the advanced or professional user (Freeplane, 2014d).

Även begreppen non-developers och non-programmers används för icke-utvecklare. Exempelvis används begreppet non-programmers när man vill påvisa att det även för icke-utvecklare går att installera paket med script, ikoner och så vidare.

Freeplane also has a powerful mechanism to install packages of scripts, icons, images, language dictionaries for spell checking and other preferences including a different menu structure; and which can be used by non-programmers. (Freeplane, 2014f)

## **Utvecklare av add-ons**

Till skillnad från de begrepp som kan sägas hänvisa till antingen icke-utvecklare eller utvecklare, kan de begrepp som används för att beskriva hur det är möjligt att göra add-ons till mjukvaran sägas riktas mot både icke-utvecklare och utvecklare.

I wikin står det att vem som helst kan skapa add-ons, vilket därmed också inkluderar icke-utvecklare.

Anybody can write an add-on! (see the add-on developer documentation) note: add-ons are not necessarily provided by the Freeplane core developers. (Freeplane, 2014b)

Detta framkommer också på wikins första-sida, där begreppet *users* används för att beskriva att det är möjligt att göra egna add-ons som sedan man sedan kan dela med sig av till andra *users*.

It is possible to define your own add-ons to Freeplane using the built in scripting language Groovy. Check page Add-ons\_(install) for apps other users have made and are sharing with you. (Freeplane, 2014f)

För att göra add-ons används dock programmeringsspråket Groovy, vilket förutsätter programmeringskunskaper som utvecklare har. Därmed kan begreppet, såsom det används av Freeplane, sägas hänvisa både till icke-utvecklare och utvecklare.

### **Sammanfattning**

För utvecklare används huvudsakligen begreppet *Developer*, vilket kan sägas benämna en programmeringskunnig person som också kan delta i allt i projektet. Det finns dock ett flertal andra begrepp som används för att beskriva samma typ av deltagare.

För icke-utvecklare kan begreppen delas upp i kategorier beroende på typ av deltagande, exempelvis tester, administrator och translator.

Vid beskrivningar om hur add-ons kan utvecklas till mjukvaran används både begrepp som kan sägas höra till utvecklare respektive icke-utvecklare, där framförallt begreppet *users* används vid beskrivningen av att alla kan skapa add-ons. Samtidigt sker utvecklingen av add-ons i programmeringsspråket Groovy, vilket kan sägas förutsätta programmeringskunskaper.

### **4.3 Beskrivning av projektet gentemot dess användargrupper**

Gentemot utvecklare beskrivs projektet främst i form av att det är enkelt att skapa och utveckla tillägg till mjukvaran och att källkoden förändrats för att underlätta för nya utvecklare. Gentemot icke-utvecklare beskrivs projektet på flera sätt. Detta sker i form av generella beskrivningar av projektet, beskrivning av mjukvarans användningsområden och funktionalitet, add-ons och scripting samt mer tekniska beskrivningar.

## Beskrivning av projektet gentemot utvecklare

Gentemot utvecklare menar man att även om funktionaliteten i Freeplane är inriktad mer mot icke-utvecklare, så är det enkelt att utveckla kraftfulla tillägg i programvaran. Man menar också att studenter genom Freeplane kan få en naturlig och stimulerande miljö att utveckla sina programmeringskunskaper.

Although Freeplane functionality is intended for non-programmers, the Freeplane project offers potential programmers a versatile scripting language. Programmers may easily develop powerful extensions and students are stimulated to learn programming in a natural and stimulating environment (Freeplane, 2014f).

Ett mål med projektet är också att förändra programvarans arkitektur så att det skall underlätta bidrag från nya utvecklare.

- current FreeMind implementation is hard to extend and maintain. I have tried to collect the most critical points here ([http://freemind.sourceforge.net/wiki/index.php/Freemind\\_refactoring\\_plan](http://freemind.sourceforge.net/wiki/index.php/Freemind_refactoring_plan)) . Such refactoring is not easy, but I think if we want to give new developer a chance to make good contributions, we must provide clear architecture and design first (Freeplane, 2011a).

För att åstadkomma detta har man målet att skapa ett stort och effektivt team med utvecklare. Man vill också att de som bidrar aktivt i projektet också skall ha mer att säga till om. Detta menar man skulle kunna innebära kortare tid mellan releaser.

- Big and effective developer Team. I believe that much more people were willing to contribute to this extraordinary sensible and helpful project as a mind map editor.
- So the development speed can be increased and time between the releases can be significantly shortened.
- People like you and me systematically contributing to the project should get more influence on the project decisions (Freeplane, 2011a).

För att underlätta för nya bidrag menar man också att det är viktigt med väldokumenterade och tydliga interface.

We strive to develop FreePlane as a lean framework for mind mapping, with clear and well-documented interfaces, allowing for easy extension and customization by programmers, scripters and other non- developers, federating them in a lively community supporting our project vision (Freeplane, 2011d).

## **Beskrivning av projektet gentemot icke-utvecklare**

Freeplane beskriver projektet gentemot icke-utvecklare på flera olika sätt. Dels i form av mer generella beskrivningar av projektet, beskrivningar av funktionalitet och mjukvarans användningsområden, add-ons och scripting samt tekniska beskrivningar.

### *Generell beskrivning av projektet*

Gentemot icke-utvecklare beskrivs projektets mål i form av bland annat bättre användbarhet tillsammans med mer funktionalitet och release oftare:

Better Mind Map editor than FreeMind

- Better usability,
- More features,
- People like you and me systematically contributing to the project should get more influence on the project decisions.
- There should be clearly communicated and broadly discussion of the further development ideas, steps, goals and visions.

A large, friendly community in which your voice will be heard (Freeplane, 2011a).

Projektet nämner också vikten av en god kommunikation gällande utvecklingen av nya idéer samt en öppen community där varje röst hörs. Programvaran å andra sidan skall kunna användas både i hemmet, skolan och på arbetet för att skapa mind maps och analysera information.

Freeplane is a free and open source software application that supports thinking, sharing information and getting things done at work, in school and at home. The software can be used for mind mapping ([https:// secure.wikimedia.org/wikipedia/en/wiki/Mind\\_map](https://secure.wikimedia.org/wikipedia/en/wiki/Mind_map)) and analyzing the information contained in mind maps (Freeplane, 2014f).

Tanken är att programvaran skall vara både snabb och enkelt att använda och vara något mellan lanting mellan en editor och ett verktyg för diagram.

Freeplane aims for maximum ease and speed of use. Occupying the middle ground between an editor and a diagramming tool, Freeplane allows the user to add content as quickly and naturally as they would in a text editor, yet producing structured content that can be manipulated as easily as a diagram. The workflow is unimpeded by the need to think about and manually place each piece of information; the user can intuitively input content as paragraphs and headings, and easily reorganise at a later stage. (Freeplane, 2014f).

### *Beskrivning av funktionalitet och användningsområden*

Bland de sidor som kan sägas vara mer inriktade mot icke-utvecklare återfinns sidor som beskriver hur programvaran kan användas för att utföra viss funktionalitet, dels hur man kan använda sig av programvaran mer generellt, exempelvis hur man kan lösa problem med hjälp av datorstöd:

Problem solving is one of the most important human activities. Here comes information about a powerful method for computer aided problem solving (Freeplane, 2012e).

Man beskriver också hur man kan använda sig av mind-maps istället för exempelvis ordbehandlare och mjukvaror som exempelvis Excel, och menar att detta är betydligt enklare att göra då man använder sig av mind-maps.

Mind Mapping Software: Mind mapping software is very flexible when it comes to processing ideas: collect thoughts and arrange them in different structures, change between detail and overview, navigate through large amounts of information, combine words and images, allow jumps from one idea to another, add tables and lengthy text notes... Try all this with usual text processors or spreadsheets – mind mapping software simply does a better job. With these ingredients, you can tackle a problem by using two mind maps: one 'tool map', prepared in advance and containing a large collection of tools, and one 'work map', where you work out a solution for a given problem. While working on the problem, you can use the tool map in different ways: ... (Freeplane, 2012e)

Projektets wiki innehåller också ett antal beskrivningar på hur funktionalitet i mjukvaran kan användas.

#### Instructions for use

- The wiki page Add-ons (install) describes how to install (and de-install) the add-on.
- Check the description of the language pack below if it replaces one or more documentation files standardly provided with Freeplane. If so, and if the add-on is of a later date than Freeplane, the following actions are possible.
  - Either you can de-install the add-on. This removes all files of the language pack.
  - Or you can remove only the older, replacing files by hand from the Freeplane user directory. Hereafter the newer files from de Freeplane download again will become active

(Freeplane, 2012a).

Dessa ser dock lite olika ut. Ovan är ett exempel på en steg-för-steg beskrivning hur man kan hämta språkpaket i form av add-ons. Andra beskrivningar är snarare beskrivna i text än i steg, vilket exempelvis syns vid beskrivningen hur objekt på mind-maps kan formateras:

Nodes can be formatted depending on its content. The rules for automatic node format-

ting consist of a condition like the conditions used for filters and a style name. Thus node formatting can depend on its content, attributes, icons or level. They can be set using "Styles->Manage conditional styles". Because each style may set only some of the formatting attributes the resulting node formatting is a sum of formats defined by explicitly set format, explicitly assigned style and all matching conditional styles (Freeplane, 2012c).

Ett annat exempel är mjukvarans sökfunktion "Approximate Search", som först beskrivs mer generellt för att sedan mer övergå till textuella steg-för-steg beskrivningar:

Starting with 1.2.13, Freeplane includes an 'Approximate Search' feature which allows to find content (nodes, notes, or whatever) that do not exactly match the search term, but allow some changes ("distance") between the search term and the content. See the section Search results below for some examples of approximate searching.

#### *How to run (approximate) Search*

Press Ctrl-F or select View->Toolbars->Filter toolbar to show the "filter/search" toolbar and enable Approximate searching by enabling the "Approximate" checkbox in the filter toolbar (Alt+A). Then you can enter a search term in the search box (to the left of Match Case) and press Enter or click the Find next Button (icon with a blue right arrow in the filter toolbar) to start the search. If you want to check for equality (search term equals content as opposed to is contained in content), then you should change the default "Contains" in the search type combobox to "Is equal to". You can define more complex search rules by using the Filter->Compose filter dialog (Freeplane, 2012b).

#### *Beskrivningar av add-ons och scripting*

I wikin jämförs add-ons till mjukvaran Freeplane med add-ons till exempelvis Mozilla Firefox och LibreOffice. En add-on i Freeplane kan sägas utöka mjukvarans funktionalitet.

Add-ons are a way to extend and customize Freeplane similar to how plug-ins ([http://en.wikipedia.org/wiki/Plug-in\\_\(computing\)](http://en.wikipedia.org/wiki/Plug-in_(computing))) and extensions ([http://en.wikipedia.org/wiki/Software\\_extension](http://en.wikipedia.org/wiki/Software_extension)) can be used to extend and customize well-known applications like Firefox (<http://addons.mozilla.org/>) or LibreOffice (<http://www.libreoffice.org/>). Freeplane add-ons can be used to provide single function, a bundle of multiple functions, bind those functions to a menu item, and much more (Freeplane, 2014b).

En del add-ons kräver att man kan skriva sina egna script, medans andra add-ons inte sägs kräva programmeringskunskaper.

There are several add-ons that don't need any programming know-how, but depending on the purpose of your add-on you might have to write scripts (Freeplane, 2014a).

Man menar att vem som helst kan göra en add-on. I wikin hänvisar man sedan till "add-on developer documentation" kring hur det man gör för att utveckla en add-on.

Anybody can write an add-on! (see the add-on developer documentation) note: add-ons are



not necessarily provided by the Freeplane core developers (Freeplane, 2014b).

För att skriva egna funktioner och script används Groovy och JavaScript, även om andra programmeringsspråk också är möjliga att använda. De funktioner eller add-ons man skapat kan man sedan dela med sig av till andra användare.

Freeplane's builtin functionality can be extended by Groovy (<http://groovy.codehaus.org/>) and JavaScript (<http://en.wikipedia.org/wiki/JavaScript>) scripts. Starting with Freeplane 1.3.5\_05 you can use many other languages, e.g Python. This page gives a first impression what you can do with Groovy scripting and helps to get started. With Freeplane scripting you can - write your own functions and use them from the menu or via keyboard shortcuts, - use formulas in your map to compute stuff like in Excel, and - create add-ons to share it with other users (Freeplane, 2014h).

Groovy beskrivs dock som ett komplett programmeringsspråk:

Groovy, the scripting language of choice in Freeplane is a full-fledged programming language which can nearly do everything with your computer if it's unconstrained. For this reason Freeplane severely restricts script execution after installation. For script development and for using scripts most of the restrictions should be disabled (Freeplane, 2013d).

För att skapa script hänvisar man också till dokumentationen av de interface som används i form av Javadoc.

For the Scripting API see its Javadoc documentation (<http://freeplane.sourceforge.net/doc/api/>) . The Scripting API is in some sense extended by Freeplane's utility classes and by the Libraries included in Freeplane (Freeplane, 2013c).

### *Tekniska beskrivningar*

Ett exempel där man beskriver en funktionalitet för icke-utvecklare och sedan dels går in i tekniska detaljer, dels förväntar sig programmeringskunskaper som enbart kan förväntas finnas hos utvecklare är när man beskriver att programvaran automatiskt kan känna av huruvida ett nummer är ett datum eller inte. Först börjar man med att beskriva funktionaliteten för icke-utvecklare:

If you set the text of a node to a number or date in Freeplane it will likely be recognized specially. If you don't like that advance to #How to switch data recognition off. The default ("Standard") number format is "#0.####", i.e. at least one digit before and at most four digits after the decimal separator (Freeplane, 2012d).

Denna formateringsfunktion kan utföras enligt wikin på två sätt, antingen likt Excel, eller ett alternativt tillvägagångssätt som bygger på en Java-funktion:

An alternative formatter ("printf") style uses the percent sign specially, see Formatter docu-

mentation (<http://docs.oracle.com/javase/1.5.0/docs/api/java/util/Formatter.html>) (again for a given input number 0.12345): (Freeplane, 2012d)

Beskrivningen av funktionaliten återgår till att fokusera på icke-utvecklare och en mindre teknisk beskrivning...

It depends on your language setting which character is used as decimal and grouping sign: In German setting "12.123" will be recognized as 12123 since in Germany "." groups Thousands while "," is used as decimal separator. In an English setting it's vice versa: "12,123" is 12123 (Freeplane, 2012d).

... för att sedan återigen ta upp det alternativa tillvägagångssätt för formatering, i form av en java-funktion. Man hänvisar också, likt ovan, till dokumentationen (javadoc) för funktionen.

A "printf" style (<http://docs.oracle.com/javase/1.5.0/docs/api/java/util/Formatter.html>) is also available for dates but more useful are the language setting depending shortcuts (Freeplane, 2012d).

## Sammanfattning

Generellt sätt är wikin uppdelad i sidor för antingen icke-utvecklare eller utvecklare, utom ett färre antal sidor som kan sägas inrikta sig gentemot både icke-utvecklare och utvecklare. Intressant är dock att man exempelvis nämner att vem som helst kan göra en add-on, där tillvägagångssättet sedan kräver en utvecklarens programmeringskunskaper. De sidor som kan sägas mer inriktade mot icke-utvecklare är också i hög grad baserade på steg-för-steg beskrivningar, även om man i många fall kommer in på något mer tekniska detaljer kring programvaran än vad som kan sägas relevant för en icke-utvecklare. Man menar också att funktionaliteten i sig är inriktad mot icke-utvecklare.

## 4.4 Möjliga sätt för deltagande i projektet

Förutom att använda sig av projektets mjukvara Freeplane beskriver man att man kan delta i projektet som utvecklare, testare, översättare och forum-administratör. Deltagande kan ske dels genom diskussionsforum, redigering av wiki, rapportering av problem och önskemål, testning av nya releaser och genom att skapa översättningar.

Join the Freeplane team not only as user but maybe as a

- developer (coder and designer)
- tester (writer of tutorials, screencasts and help texts)
- translator (various languages)
- administrator (forum moderator)

Welcome! (Freeplane, 2014f)

Projektets mål är att få ett trevligt klimat där inga bidrag kastas bort, under förutsättning att dessa inte helt avviker från utvecklarnas visioner. De bidrag som görs skall också snabbt komma med i nästa release av mjukvaran.

It was felt that the current FreeMind set up was not doing enough both to promote and integrate community contributions. Freeplane aims to create a contribution friendly climate. We hope to ensure that contributions do not go to waste. If you can contribute, and your ideas are not wildly out of sync with the vision of the developers, then your contribution will be rapidly integrated and included in the new release (Freeplane, 2011c).

I projektets vision framkommer också att alla i projektet skall ha frihet att utveckla och dokumentera sina idéer och ges chansen att övertyga community och utvecklare att implementera dessa idéer i projektet.

As a project member anyone has the freedom to develop his own ideas and document them, to implement them or have a chance to inspire and convince the community and the coders to implement the ideas (Freeplane, 2011d).

### **Deltagande för utvecklare**

Projektet menar att man erbjuder utvecklare en god miljö för att bygga ut Freeplane och att Freeplane också kan ses som en bra miljö för studenter att lära sig programmering.

Although Freeplane functionality is intended for non-programmers, the Freeplane project offers potential programmers a versatile scripting language. Programmers may easily develop powerful extensions and students are stimulated to learn programming in a natural and stimulating environment. (Freeplane, 2014f).

Källkoden, som bygger på projektet FreeMind, har gjorts om så att denna skall bli lättare så att detta skall underlätta för nya utvecklare att delta i utvecklingen. Man menar också att detta skall göra det möjligt att släppa nya versioner oftare.

Freeplane has been refactored from Freemind to make its code and packages more modular, make it easier for new software developers to contribute to Freeplane development, and to

release new versions of Freeplane more often. (Freeplane, 2014f).

Diskussionen kring utvecklingen sker framförallt i ett diskussionsforum för utvecklarna samt en e-postlista.

- In the Developer discussion forum and mailing list (<http://freeplane-developer.996965.n3.nabble.com/>) developers are encouraged to discuss feature development (HOW) (Freeplane, 2014f).

I projektet används också ett distribuerat versionshanteringssystem, GIT, för att hantera källkoden.

Freeplane uses a distributed revision control ([http://en.wikipedia.org/wiki/Distributed\\_revision\\_control](http://en.wikipedia.org/wiki/Distributed_revision_control)) system named git. Use this instead of the source packages that are contained in the releases if you plan to take up coding. Our master repository is available at <https://github.com/freeplane/freeplane>  
- Learn how to use git and get the code (Freeplane, 2014e).

### **Deltagande för icke-utvecklare**

Icke-utvecklaren kan delta i projektet genom diskussionsforum, skapa add-ons, bidra till wiki, rapportera problem och önskemål, testa nya releaser och bidra med översättningar.

#### *Delta i diskussionsforum*

För diskussion kring utvecklingen gentemot icke-utvecklare hänvisar man till ett öppet forum för frågor och diskussion.

User forum Open forum Freeplane (<https://sourceforge.net/apps/phpbb/freeplane/viewforum.php?f=1>) for questions and discussions. (Freeplane, 2014d)

På projektets startsida beskrivs detta forum som ett forum för användare att diskutera befintlig funktionalitet.

- In the Open discussion forum (<https://sourceforge.net/p/freeplane/discussion/758437>) users mainly discuss the usage of existing features (WHAT) (Freeplane, 2014f).

#### *Skapa add-ons*

I wikin beskriver man att alla kan skapa add-ons för Freeplane och att detta inte enbart är något som kan tillhandahållas av utvecklarna av Freeplane.

Anybody can write an add-on! (see the add-on developer documentation) note: add-ons are not necessarily provided by the Freeplane core developers (Freeplane, 2014b)

De add-ons som utvecklas kan sedan delas med andra. Att göra add-ons är inte bara möjligt för alla, utan man menar att detta också är lättare än man tror. Scripten skrivs i Groovy.

It is possible to define your own add-ons to Freeplane using the built in scripting language Groovy. Check page Add-ons\_(install) for apps other users have made and are sharing with you. They can be downloaded and installed with a simple mouse click. Instructions are available if you want to make your own addons. With Freeplane and groovy it is more easy than you think. (Freeplane, 2014f).

De script man gjort kan laddas upp på wikin. Diskussion om script-funktionalitet som exempelvis saknas rapporteras till diskussionsforumet som inriktar sig mot icke-utvecklare. Man hänvisar också till en bugtracker.

Wanted: Your participation! It's very likely that scripting support lacking some functionality that would be useful for a large number of users. For this reason you are strongly encouraged to give feedback on issues you are having with scripting and on things you are missing. - For discussions use the Freeplane open discussion forum (<http://sourceforge.net/apps/phpbb/freeplane/viewforum.php?f=1>) . - For bugs and feature requests use the Mantis issue tracker (<https://sourceforge.net/apps/mantisbt/freeplane>) . - Please add useful scripts to the Scripts collection Wiki page. - To ask questions directly related to this page, use the discussion page. (Freeplane, 2014h).

För att underlätta vidareutveckling av add-ons kan man enligt projektet använda sig av system för versionskontroll. Något som kan sägas kräva programmeringskunskaper.

To facilitate collaboration and continuous improvement put the source-code under version control in a public Github, Bitbucket or SourceForge repository. If you want you could contribute to the central Freeplane add-on repository (<http://github.com/freeplane/addons>) (either as a Freeplane developer or via pull requests (<http://github.com/freeplane/addons/pulls>)). In most cases you will have your own repository (Freeplane, 2013a).

### *Bidra till wiki*

Icke-utvecklare bjuds in till att delta genom att redigera wiki-sidor. Dessa är dock låsta, vilket innebär att man måste bli en registrerad editor i wikin för att göra detta. Det är möjligt att dela med sig av tips och information genom att bidra till wikin.

You are invited to use and contribute to this wiki, to share information and tips about Freeplane and its features. NOTE: because of spammer attacks, editing of this wiki is now only allowed for registered wiki editors. If you want to become a wiki editor, write (<http://sourceforge.net/apps/phpbb/freeplane/viewforum.php?f=1>) to the Freeplane team. (Freeplane, 2014f)

Att få rättigheter att bidra menar man dock inte är något problem, utan alla kan få rättigheter

för detta. I wikin kan icke-utvecklaren också dela med sig av mindmaps till andra användare.

How to contribute new maps You'll need to be a registered user of this wiki with Administrator rights to be able to publish mind maps to the gallery. Don't let that deter you, anyone can be granted Administrator rights. (Freeplane, 2014g)

### *Rapportera problem och önskemål*

Problem och önskemål kan dels diskuteras i projektets forum, där man menar att man har en aktiv community till vilken man också kan ställa frågor.

Open forums give fast and easy access to the active user community of Freeplane. You can ask questions or post suggestions. (Freeplane, 2014i)

Buggar och önskemål om nya funktionalitet kan också lämnas i en bugtracker.

We invite you to ask for new features or report bugs using our issue tracker ([https://sourceforge.net/apps/mantisbt/freeplane/view\\_all\\_bug\\_page.php](https://sourceforge.net/apps/mantisbt/freeplane/view_all_bug_page.php)) (Freeplane, 2014f).

### *Testa nya releaser*

Icke-utvecklare kan också delta genom att prova nya versioner av mjukvaran. Förutom att prova beta-versionen av mjukvaran kan man också prova preview-versioner.

Freeplane is being developed fast. If you want to use the newest functionality consider using a beta version. Besides you are invited to take part in testing the preview versions. (Freeplane, 2014i)

Då nya versioner testas och man skall rapportera problem via projektets bugtracker, uppges ett antal saker som skall vara med vid rapporteringen. Till dessa hör exempelvis en beskrivning vad man gjort då felet uppkommit, felmeddelanden, version av mjukvara, operativsystem och vad som var det önskade resultatet.

Testing an official release 3. Report any problems via the bug tracker (<https://sourceforge.net/apps/mantisbt/freeplane/>) (under Project->Bugs). Please include as much information as possible:

- what you were doing
- the behavior you encountered
- error or warning messages you got
- your Freeplane and Java versions
- your operating system

ALWAYS SUBMIT:

- a description of steps to reproduce the bug
- the actual results of following these steps
- the expected / desired results (Freeplane, 2011b)

### *Bidra med översättningar*

Freeplane finns översatt till ett flertal språk. Det är möjligt att delta genom att göra en översättning till sitt eget språk, diskussion kring detta sker i forumet. Felaktiga översättningar rapporteras i projektets bugtracker.

Some kind people have decided to translate Freeplane in their own language. You can report translation bugs as regular issues in Mantis Bug Tracking. If you want to have discussions about translation in our native tongue, you can find here the list of forums topics related to translations. (Freeplane, 2013b)

### **Sammanfattning**

Freeplane menar att man erbjuder en god miljö för att skapa tillägg och lära sig programmera. För att delta kan utvecklaren delta i diskussionsforum och e-postlista. Icke-utvecklare å andra sidan kan delta i projektet genom att skapa add-ons, bidra till wiki, rapportera problem och önskemål, testa nya releaser, delta i diskussionsforum och bidra med översättningar. Projektet har ett mål som beskriver att inga bidrag skall kastas bort om dessa inte strider mot utvecklarnas visioner, och projektets vision framhåller att alla i projektet skall ha frihet att utveckla sina idéer.

## 5 Analys

I detta kapitel redogörs för studiens analys.

### 5.1 Utvecklingsprocessen

Projektet Freeplane kan sägas vara ett open source software-initiativ, vilket innebär att detta baseras på att frivilliga tillsammans utvecklar en mjukvara där en ägare påbörjat projektet, vilken också har rätt att distribuera nya versioner av mjukvaran. Om de bidrag som skickas in, exempelvis källkod, är tillräckligt bra accepteras dessa i projektet och deltagaren får äran för bidraget (Oreg & Nov, 2008). I projektet Freeplane menar man att alla bidrag ska accepteras, så länge dessa inte strider mot utvecklarnas visioner (Freeplane, 2011c). Detta kan tolkas på flera sätt: antingen att utvecklaren godtyckligt kan välja att acceptera vissa bidrag framför andra, vilket innebär att utvecklaren kan sägas ha en större makt än icke-utvecklarna gällande vad som kommer med i projektet. En annan tolkning kan vara att utvecklaren skall acceptera alla bidrag utan någon större utsortering, detta då det i projektets vision inte framkommer vad som är ett "tillräckligt bra" bidrag. Att alla bidrag skall tas med, kan också tolkas som att även icke-utvecklarnas bidrag är lika viktiga.

Bruns (2012) beskriver att kollaborativt skapande av innehåll genom internet-communitys exempelvis lett till användbara resurser i form av open source-mjukvara. Freeplane kan sägas vara ett exempel på detta, där utvecklingen sker på frivillig basis och mjukvaran släpps under licensformen GPL, vilken uppfyller Stallmans (2002) mening att alla ska få tillgång till ett fritt fungerande system som är möjligt att köra, kopiera, distribuera, förändra och förbättra.

Två sätt att beskriva utvecklingen av fri programvara och open source, är katedralen och basaren. I katedralen sker utvecklingen av ett fåtal utvecklare och källkoden släpps samtidigt som en ny version släpps. I basarmodellen är å andra sidan källkoden hela tiden tillgänglig, och utvecklingen sker öppet via internet (Raymond, 2001). Projektet Freeplane kan jämföras med basarmodellen, där källkoden hela tiden är tillgänglig över internet, med hjälp av versionshanteringssystemet GIT. Freeplane kan dock sägas ha ett begränsat antal deltagare, åtminstone enligt projektets lista över bidragsgivare (Freeplane, 2014c). Detta kan bero på två saker: antingen har projektet ett begränsat antal deltagare, eller så har man många deltagare men att dessa inte ges äran för bidraget. Åtminstone inte i listan över bidragsgivare, vilket kanske är mer troligt då



man uppger att man har en aktiv community (Freeplane, 2014f).

## 5.2 Projektets community

Bruns (2012) menar att deltagare i open source-projekt skapar mer eller mindre tydliga strukturer. När dessa blir mer permanenta benämns dessa communitys. Dessa existerar sedan kring en kärna av aktiva och engagerade deltagare med gemensamma praktiker och värderingar.

Freeplane beskriver att de har en aktiv community och projektet lägger vikt vid deltagande i utvecklingsprocessen (Freeplane, 2014f). Målet är att skapa en bra programvara genom support till varandra i communityn och god kommunikation gällande utvecklingen av nya idéer. Man påpekar också vikten av en öppen community, där varje röst hörs, samt att denna skall vara vänligt inställd. I beskrivningen av projekts mål framkommer också att inga bidrag skall kastas bort (Freeplane, 2011a). I projekts vision menar man att alla i projektet skall ha frihet att utveckla och dokumentera sina idéer och ges chansen att övertyga community och utvecklare att implementera idéerna (Freeplane, 2011d). Att man i projektet så tydligt påpekar vikten av en öppen community, att ge support till varandra, att alla röster skall höras och att alla bidrag skall accepteras så länge dessa inte går emot utvecklarnas visioner, tyder på att projektets kärna av utvecklare utvecklat gemensamma praktiker och värderingar likt Bruns (2012) beskrivning av communitys.

## 5.3 Icke-utvecklare och utvecklare

Inom deltagande design är det viktigt att de som skall använda den informationsteknik som utvecklas också kan delta i den kollaborativa utvecklingsprocessen. Detta är särskilt tydligt i den skandinaviska traditionen i deltagande design, där särskild vikt läggs vid att de som skall använda informationstekniken också har en central roll i dess design (Simonsen & Robertson, 2013). En uppdelning mellan användare i form av utvecklare och icke-utvecklare kan också göras, där utvecklare har programmeringskunskaper och är utvecklare i projektet, till skillnad från icke-utvecklare som inte vill delta i programmeringen av den mjukvara som utvecklas (Iivari, 2011).

Projekts designartifakt, Freeplane, skall vara användarvänlig, snabb och enkel att använda och gå att använda i hemmet, i skolan och på arbetet. Funktionaliteten sägs vara inriktad mot icke-

utvecklare (Freeplane, 2014f) och ökad användbarhet anges överst i projektets mål (Freeplane, 2011a). Således kan icke-utvecklare sägas vara de som skall använda sig av den informationsteknik som utvecklas.

I projektet Freeplane benämns utvecklare oftast Developer och kan sägas ha programmeringskunskaper och kunna delta i allt i projektet. Icke-utvecklare, vilka inte har programmeringskunskaper och tekniska kunnande benämns efter vilken typ av deltagande som sker - exempelvis testare, översättare och administratör (för forum och/eller wiki). I projektet Freeplane återfinns en tredje grupp användare som kan sägas vara ett mellanting mellan utvecklare och icke-utvecklare, de som utvecklar add-ons. Detta är något som beskrivs som att alla kan göra, men beskrivningarna för tillvägagångssättet förutsätter programmeringskunskaper.

## 5.4 Deltagande

För att få icke-utvecklarnas röster hörda i utvecklingsprocessen föreslår Simonsen och Robertson (2013) att man använder sig av verktyg för att underlätta detta, exempelvis mock-ups och prototyper. Utvecklingsprocessen måste också inkludera ett utbyte av kunskap och gemensamt lärande.

Som nämnts ovan så kan deltagandet i Freeplane exempelvis ske genom att delta i discussionsforum, redigera wiki-sidor, rapportera problem och önskemål, testa nya releaser eller att skapa översättningar. I projektet verkar dock inte deltagande i form av mock-ups och prototyper förekomma, åtminstone inte kontinuerligt i utvecklingsprocessen.

Inom deltagande design finns det politiska och pragmatiska aspekter gällande deltagande i utvecklingsprocessen. De pragmatiska handlar bland annat om att det är viktigt att utvecklare och icke-utvecklare lär sig tillsammans, vilket är en central del i hela utvecklingsprocessen (Simonsen & Robertson, 2013). När det gäller de politiska aspekterna handlar dessa främst om att alla röster skall få höras, vilket av Simonsen och Robertson (2013) beskrivs som en grundläggande rättighet för icke-utvecklaren att påverka designen av informationstekniken denne skall komma att använda.

Gällande kontinuerligt lärande mellan icke-utvecklare och utvecklare så har det inte framkommit någonting vid undersökningen av projektets wiki som tyder på att icke-utvecklarna hela tiden varit delaktiga i utvecklingsprocessen. Snarare är projektet initierat av en projektägare,

benämnd *founder* i projektet (Freeplane, 2014c), likt Oreg och Novs (2008) beskrivning av open source-software initiativ ovan.

Att alla röster skall få höras är någonting som framhålls mycket i projektet, bland annat i projektets mål. Man menar att man har en aktiv community i vilken man skall ge support till varandra (Freeplane, 2014f). Freeplanes community skall också vara vänligt inställd och varje röst skall höras (Freeplane, 2011a). I projektets vision beskriver man att alla i projektet skall ges möjlighet att dokumentera och utveckla sina idéer (Freeplane, 2011d).

Å andra sidan beskriver man att inga bidrag skall kastas bort, så länge dessa inte avviker från utvecklarnas visioner (Freeplane, 2011c). I projektets mål beskriver man också att de som bidrar mest aktivt i projektet också skall vara de som har mest att säga till om (Freeplane, 2011a).

#### **5.4.1 Direkt och indirekt deltagande**

Deltagande i utvecklingsprocessen kan ske genom att icke-utvecklaren själv deltar direkt, eller att denne deltar indirekt via representanter (Iivari, 2011). I de beskrivningar som finns om deltagande i projektet Freeplane har det inte framkommit deltagande i form av representanter, utan deltagandet kan ske i form av att icke-utvecklaren själv deltar.

#### **5.4.2 Roller vid deltagandet**

Inom open source där deltagande design kan sägas ske distribuerat, handlar detta om att förstå icke-utvecklarnas arbetssätt, vilket förändras tillsammans med dessa samt samla in feedback från icke-utvecklarna för att tillsammans skapa en lösning. Målet är att förbättra funktionalitet eller användbarhet (Iivari, 2011).

I projektet Freeplane omnämns dessa två punkter, ökad användbarhet och funktionalitet, i listan över projektets mål (Freeplane, 2011a). Gällande icke-utvecklarens roll vid deltagande i projektet så kan den delta på ett antal olika sätt, exempelvis i form av testare, översättare och administratör (för forum och wiki).

## 5.5 Möjliga sätt för icke-utvecklare att delta i projektet

Utvecklingen inom open source sker distribuerat både geografiskt och organisationsmässigt och i tiden, vilket gör att deltagandet blir beroende av kommunikation och samarbete över internet. Detta kan ske genom exempelvis e-postlistor, diskussionsforum, versionshanteringssystem och buggrapporteringsverktyg (Iivari, 2011). I Freeplane kan deltagandet ske på flera sätt för icke-utvecklare.

Icke-utvecklarna kan delta i diskussionsforum, där dessa hänvisas till ett öppet forum där nuvarande funktionalitet diskuteras. Detta till skillnad från utvecklare som hänvisas till ett utvecklarforum och en e-postlista där ny funktionalitet diskuteras (Freeplane, 2014f).

I projektet menar man att alla kan skapa add-ons (Freeplane, 2014b), således gäller detta också icke-utvecklare. Detta sker sedan i programmeringsspråket Groovy, vilket förutsätter programmeringskunskaper, även om manar att det är enkelt (Freeplane, 2014f).

Icke-utvecklare kan också delta i projektet genom att redigera wiki-sidor i projektets wiki, rapportera problem, önskemål och buggar i projektets buggrapporteringsverktyg (Freeplane, 2014f). Andra möjliga sätt att delta för icke-utvecklare är att testa nya versioner av mjukvaran (Freeplane, 2014i) och bidra med översättningar (Freeplane, 2013b).

## 6 Slutsats och diskussion

Freeplane är ett traditionellt open source-projekt som syftar till deltagande, vilket syns då projektet har som mål att skapa en väl fungerande mjukvara genom support inom communityn och god kommunikation vid utvecklingen av nya idéer. Projektet kan ses som ett open source software-initiativ, vilket beskrivs av Oreg och Nov (2008) som att frivilliga i en organisation tillsammans utvecklar en mjukvara.

Alla deltagare i projektet skall ges möjlighet att utveckla och dokumentera sina idéer och bidrag skall inte kastas bort i onödan. Man påpekar också vikten av att varje röst hörs, att ha en öppen community och att denna skall vara vänligt inställd. Detta kan jämföras med deltagande design, vilket bland annat har till avsikt att få icke-utvecklarnas röst hörd i utvecklingsprocessen (Simonsen & Robertson, 2013).

I deltagande design är delaktighet central i flera steg i utvecklingsprocessen, exempelvis då användbarheten i en mjukvara skall förbättras (Hess m. fl., 2013). Detta kan relateras till projektet Freeplane och vars designartifakt, mjukvaran Freeplane, skall vara användarvänlig och enkel att använda. Funktionaliteten inriktar sig mot icke-utvecklare och ökad användbarhet anges överst i projektets mål.

Även om man i projektet beskriver vikten av en öppen community där varje röst hörs och att alla skall ha frihet att utveckla sina idéer, sker ändå ett visst fokus på utvecklaren i utvecklingsprocessen. Exempelvis på följande sätt:

- Användare hänvisas till att diskutera nuvarande funktionalitet i ”Open discussion forum”. Ny funktionalitet diskuteras i ”Developer discussion forum”, dit utvecklare hänvisas att delta.
- Projektet skall ha ett trevligt klimat där inga bidrag kastas bort, under förutsättning att dessa inte avviker för mycket från utvecklarnas visioner.
- De som kontinuerligt bidrar skall också ha en större bestämmanderätt när det gäller beslut i projektet.
- Beskrivningar av funktionalitet innehåller tekniska beskrivningar som kan förväntas förstås av utvecklare, men inte nödvändigtvis av icke-utvecklare.
- Man menar att alla på ett enkelt sätt göra egna tillägg, vilket sedan enligt beskrivningarna

görs i programmeringsspråket Groovy.

- Ett stort mål med projektet är att förändra av mjukvarans arkitektur.
- Ökad användbarhet är listat överst i projektets mål, tillsammans med utökad funktionalitet. Detta menar man skall åstadkommas genom ett stort och effektivt utvecklarteam.

Därmed har projektet ett utvecklarperspektiv likt Stallman (2002) och Raymond (2001), vilket kan återfinnas i många traditionella open source-projekt. Framförallt handlar detta utvecklarperspektiv i Freeplane om att icke-utvecklare hänvisas till ett forum där deltagande av befintlig funktionalitet sker, och utvecklare till ett annat där diskussion om ny funktionalitet sker. Här inbjuder således inte projektet icke-utvecklarna till diskussion om ny funktionalitet i mjukvaran som utvecklas. Man menar också att bidrag till projektet skall inte heller kastas bort, men detta gäller bara under förutsättning att de inte avviker från utvecklarnas visioner. Detta kan tolkas som att utvecklaren därmed har mer att säga till om i utvecklingsprocessen. I textanalysen framkom det även att man i projektet menar att de som bidrar kontinuerligt i projektet skall ha mer att säga till om. Då detta troligen är projektets kärna av utvecklare, så kan även detta ses som att utvecklarna skall ha mer bestämmanderätt än icke-utvecklarna. Beskrivningar av funktionalitet i designartifakten innehåller också tekniska beskrivningar, vilka inte alltid kan förväntas förstås av icke-utvecklare, detta då icke-utvecklare inte kan förväntas ha samma tekniska kunnande som utvecklare. På ett liknande sätt förutsätts programmeringskunskaper vid beskrivningar gällande att alla kan göra add-ons, vilket sedan görs i programmeringsspråket Groovy. Ett stort mål med projektet är också att förändra mjukvarans arkitektur för att underlätta deltagande. Att mjukvarans arkitektur förändras är inte nödvändigtvis av ondo, men en sådan förändring underlättar enbart för utvecklare, inte icke-utvecklare. Ökad användbarhet är ett annat stort mål, vilket man avser åstadkomma genom bland annat ett stort och effektivt utvecklarteam.

Iivari (2011) beskriver att deltagande antingen kan ske direkt eller indirekt genom representanter för användarna. I projektet Freeplane kan icke-utvecklare bland annat delta i rollen som testare, översättare och administratör för wiki eller forum. I studiens textanalys framkommer inga representanter för deltagare i projektet.

Det är viktigt att uppmuntra icke-utvecklare att spontant och frivilligt delta genom att bidra med exempelvis feedback, vilket förutsätter en aktiv community och ett forum (Iivari, 2011). Projektet menar att man har en aktiv community, det finns också ett forum för deltagande. Detta är

dock uppdelat i två delar, ett som inriktar sig mot icke-utvecklare där den nuvarande funktionaliteten diskuteras, och ett som inriktar sig mot utvecklare där ny funktionalitet diskuteras.

Det kan vara svårt att kommunicera användbarhetsproblem textuellt (Iivari, 2011). För att diskutera designfrågor snabbt och publikt i projekt där deltagarna befinner sig på olika geografiska platser, vilket sker i Freeplane, har bloggar föreslagits som ett alternativ. I analysen av Freeplane framkommer inga verktyg som används för att kommunicera användbarhetsproblem.

Sammanfattningsvis kan Freeplane sägas vara ett projekt som från början vill uppmuntra icke-utvecklarnas deltagande i utvecklingsprocessen. I traditionella open source-projekt är det dock vanligt med ett utvecklarperspektiv, likt det som återfinns i Stallmans (2002) och Raymonds (2001) beskrivningar av open source och fri programvara. Detta utvecklarperspektiv återfinns även hos projektet Freeplane, vilket kan sägas begränsa möjligheterna för icke-utvecklarnas deltagande i utvecklingsprocessen. Inom deltagande design menar man att deltagandet av icke-utvecklarna helst skall vara kontinuerligt, vilket dock är svårt att uppnå (Hess m. fl., 2013). Projektet Freeplane kan sägas initierat av en utvecklare som också kan sägas inneha rollen som projektägare. Genom att få icke-utvecklarna delaktiga redan från början eller åtminstone tidigare i utvecklingen, kanske detta och andra traditionella open source-projekt skulle kunna få ett minskat fokus på utvecklaren och ge icke-utvecklarna mer att säga till om i utvecklingsprocessen.

## 6.1 Metoddiskussion

Studiens syfte är att undersöka hur riktlinjer i traditionella open source-projekt beskriver att icke-utvecklare kan delta i utvecklingsprocessen. För att undersöka detta har jag valt en kvalitativ forskningsstrategi och gjort en kvalitativ textanalys av projektet Freeplanes samtliga wikisidor. Fördelen med att göra en kvalitativ textanalys av projektets wiki, framför exempelvis intervjuer och observationer, är att en textanalys av dessa texter och riktlinjer kan lyfta fram underliggande synsätt och avspegla föreställningar i den miljö där texten skrivs, i det här fallet projektet Freeplane. Freeplanes wiki fungerar också som hemsida för projektet och kan därmed vara en av de första kontakterna icke-utvecklare får med projektet. Detta kan i sin tur få konsekvenser för hur icke-utvecklare väljer att delta.

En begränsning med detta tillvägagångssätt är att enbart projektets beskrivning av deltagande

undersöks. Därmed går det inte att säga hur de som deltar sedan deltar i praktiken. För att göra detta hade istället intervjuer eller observationer varit mer lämpliga. Med exempelvis intervjuer är det istället möjligt att ställa frågor kring hur icke-utvecklare deltar, hur man uppfattar att man får delta, varför man vill eller inte vill delta. Genom intervjuer är det möjligt att ta reda på uppfattningar, värderingar, kunskaper och attityder kring deltagande.

I denna studie menar jag att projektet Freeplane har ett utvecklarperspektiv. För att göra en jämförelse om hur riktlinjer i traditionella open source-projekt beskriver att deltagande kan ske och om fler projekt har liknande riktlinjer, skulle en kvantitativ innehållsanalys kunna göras. Andra alternativa tillvägagångssätt hade varit att göra en deltagande observation, vilket kan ge ett inifrån-perspektiv. Detta skulle kunna användas för att hitta faktorer som påverkar deltagandet i projektet, exempelvis kulturella aspekter i projektet, vilka inte är tillgängliga för de som ännu inte deltar.

## **6.2 Fortsatt forskning**

Denna studie har genom en textanalys undersökt hur riktlinjer i ett traditionellt open source-projekt beskriver att icke-utvecklare kan delta i utvecklingsprocessen. Ett förslag till vidare forskning är att göra en liknande undersökning men på fler projekt traditionella open source-projekt, exempelvis genom kvantitativ innehållsanalys, för att se om fler projekt som syftar till deltagande hos icke-utvecklare också har ett utvecklarperspektiv. Ett annat alternativ till vidare studie är att utföra intervjuer i liknande projekt för att förstå hur de utvecklare i kärnan av traditionella open source-projekt ser på deltagande och vilken bakgrund dessa har, i jämförelse med till exempel icke-utvecklare som deltar i projekten.



## 7 Referenser

- Andreasen, M. S., Nielsen, H. V., Schrøder, S. O. & Stage, J. (2006). Usability in open source software development: Opinions and practice. *Information Technology and Control*, 35(3A), 303-312.
- Bergström, G. & Boréus, K. (red.). (2012). *Textens mening och makt*. Studentlitteratur.
- Bruns, A. (2012). Reconciling community and commerce? *Information, Communication & Society*, 15(6), 815-835. Hämtad 2014-03-07, från <http://www.tandfonline.com/doi/abs/10.1080/1369118X.2012.680482>
- Esaiasson, P., Gilljam, M., Oscarsson, H. & Wängnerud, L. (2007). *Metodpraktikan: konsten att studera samhälle, individ och marknad*. Norstedts Juridik.
- Freeplane. (2011a). *Goals*. Hämtad 2014-07-30, från <http://freeplane.sourceforge.net/wiki/index.php/Goals>
- Freeplane. (2011b). *How to test*. Hämtad 2014-07-30, från [http://freeplane.sourceforge.net/wiki/index.php/How\\_to\\_test](http://freeplane.sourceforge.net/wiki/index.php/How_to_test)
- Freeplane. (2011c). *Relationship to freemind*. Hämtad 2014-07-30, från [http://freeplane.sourceforge.net/wiki/index.php/Relationship\\_to\\_FreeMind](http://freeplane.sourceforge.net/wiki/index.php/Relationship_to_FreeMind)
- Freeplane. (2011d). *Visions*. Hämtad 2014-07-30, från <http://freeplane.sourceforge.net/wiki/index.php/Visions>
- Freeplane. (2012a). *Add-ons (language pack)*. Hämtad 2014-07-30, från [http://freeplane.sourceforge.net/wiki/index.php/Add-ons\\_\(Language\\_pack\)](http://freeplane.sourceforge.net/wiki/index.php/Add-ons_(Language_pack))
- Freeplane. (2012b). *Approximate search*. Hämtad 2014-07-30, från [http://freeplane.sourceforge.net/wiki/index.php/Approximate\\_search](http://freeplane.sourceforge.net/wiki/index.php/Approximate_search)
- Freeplane. (2012c). *Conditional node styles*. Hämtad 2014-07-30, från [http://freeplane.sourceforge.net/wiki/index.php/Conditional\\_node\\_styles](http://freeplane.sourceforge.net/wiki/index.php/Conditional_node_styles)
- Freeplane. (2012d). *Data recognition and data formats*. Hämtad 2014-07-30, från [http://freeplane.sourceforge.net/wiki/index.php/Data\\_recognition\\_and\\_data\\_formats](http://freeplane.sourceforge.net/wiki/index.php/Data_recognition_and_data_formats)

Freeplane. (2012e). *Freeplane and problem solving*. Hämtad 2014-07-30, från [http://freeplane.sourceforge.net/wiki/index.php/Freeplane\\_and\\_problem\\_solving](http://freeplane.sourceforge.net/wiki/index.php/Freeplane_and_problem_solving)

Freeplane. (2013a). *How to publish add-ons*. Hämtad 2014-07-30, från [http://freeplane.sourceforge.net/wiki/index.php/How\\_to\\_publish\\_add-ons](http://freeplane.sourceforge.net/wiki/index.php/How_to_publish_add-ons)

Freeplane. (2013b). *More about freeplane translation*. Hämtad 2014-07-30, från [http://freeplane.sourceforge.net/wiki/index.php/More\\_about\\_Freeplane\\_translation](http://freeplane.sourceforge.net/wiki/index.php/More_about_Freeplane_translation)

Freeplane. (2013c). *Scripting api*. Hämtad 2014-07-30, från [http://freeplane.sourceforge.net/wiki/index.php/Scripting\\_API](http://freeplane.sourceforge.net/wiki/index.php/Scripting_API)

Freeplane. (2013d). *Scripting: Security considerations*. Hämtad 2014-07-30, från [http://freeplane.sourceforge.net/wiki/index.php/Scripting:\\_Security\\_considerations](http://freeplane.sourceforge.net/wiki/index.php/Scripting:_Security_considerations)

Freeplane. (2014a). *Add-ons (develop)*. Hämtad 2014-07-30, från [http://freeplane.sourceforge.net/wiki/index.php/Add-ons\\_\(Develop\)](http://freeplane.sourceforge.net/wiki/index.php/Add-ons_(Develop))

Freeplane. (2014b). *Add-ons (install)*. Hämtad 2014-07-30, från [http://freeplane.sourceforge.net/wiki/index.php/Add-ons\\_\(install\)](http://freeplane.sourceforge.net/wiki/index.php/Add-ons_(install))

Freeplane. (2014c). *Contributors*. Hämtad 2014-07-30, från <http://freeplane.sourceforge.net/wiki/index.php/Contributors>

Freeplane. (2014d). *Freeplane tutorial extensions*. Hämtad 2014-07-30, från [http://freeplane.sourceforge.net/wiki/index.php/Freeplane\\_Tutorial\\_Extensions](http://freeplane.sourceforge.net/wiki/index.php/Freeplane_Tutorial_Extensions)

Freeplane. (2014e). *Get the code*. Hämtad 2014-07-30, från [http://freeplane.sourceforge.net/wiki/index.php/Get\\_the\\_code](http://freeplane.sourceforge.net/wiki/index.php/Get_the_code)

Freeplane. (2014f). *Home*. Hämtad 2014-07-30, från <http://freeplane.sourceforge.net/wiki/index.php/Home>

Freeplane. (2014g). *Mind map gallery*. Hämtad 2014-07-30, från [http://freeplane.sourceforge.net/wiki/index.php/Mind\\_map\\_gallery](http://freeplane.sourceforge.net/wiki/index.php/Mind_map_gallery)

Freeplane. (2014h). *Scripting*. Hämtad 2014-07-30, från <http://freeplane.sourceforge.net/wiki/index.php/Scripting>

Freeplane. (2014i). *Support*. Hämtad 2014-07-30, från <http://freeplane.sourceforge.net/wiki/index.php/Support>

Goldman, R. & Gabriel, R. P. (2005). *Innovation happens elsewhere: Open source as a business strategy*. Morgan Kaufmann Publishers.

Hess, J., Randall, D., Pipek, V. & Wulf, V. (2013). Involving users in the wild - participatory product development in and with online communities. *International Journal of Human-Computer Studies*, 71(5), 570 - 589. Hämtad från <http://www.sciencedirect.com/science/article/pii/S1071581913000074>

Iivari, N. (2011). Participatory design in oss development: interpretive case studies in company and community oss development contexts. *Behaviour & Information Technology*, 30(3), 309-323. Hämtad från <http://www.tandfonline.com/doi/abs/10.1080/0144929X.2010.503351>

Oreg, S. & Nov, O. (2008). Exploring motivations for contributing to open source initiatives: The roles of contribution context and personal values. *Computers in Human Behavior*, 24(5), 2055 - 2073. Hämtad från <http://www.sciencedirect.com/science/article/pii/S0747563207001537>

Patel, R. & Davidson, B. (2011). *Forskningsmetodikens grunder*. Studentlitteratur.

Raymond, E. S. (2001). *Katedralen och basaren: En oavsiktlig revolutionärs tankar kring linux och öppen källkod* (R. Sjölander, övers.). Nya Doxa.

Simonsen, J. & Robertson, T. (red.). (2013). *Routledge international handbook of participatory design*. Routledge International Handbooks.

Sourceforge. (2014). *About*. Hämtad 2014-03-28, från <http://sourceforge.net/about>

Stalder, F. (2005). *Open cultures and the nature of networks*. Futura publikacije.

Stallman, R. M. (2002). *Free software free society: selected essays of Richard M. Stallman* (J. Gay, red.). Free Software Foundation.