Improving Market Adoption of Mukumba

A study of techniques
for promoting open source software

CHRISTINA NILSSON

KTH Computer Science
and Communication

Bachelor of Science Thesis
Stockholm, Sweden 2011

Improving Market Adoption of Mukumba

A study of techniques
for promoting open source software

C H R I S T I N A N I L S S O N

Bachelor's Thesis in Engineering and Management (15 ECTS credits)
at the School of Industrial Engineering and Management
Royal Institute of Technology year 2011
Supervisor at CSC was Cristian Bogdan
Examiner was Stefan Arnborg

URL: www.csc.kth.se/utbildning/kandidatexjobb/teknikmanagement/2011/nilsson_christina_K11100.pdf

Royal Institute of Technology
School of Computer Science and Communication

KTH CSC
SE-100 44 Stockholm, Sweden

URL: www.kth.se/csc

Abstract

A web development framework is a foundation for web applications containing standard functionality normally used in web applications. This thesis investigates why Makumba, an open source web development framework, designed particularly for providing a short learning curve, has not gained the expected user basis, and how the Makumba community can act to facilitate market adoption of its project. This is done through studies of classic marketing theory along with case studies of a number of successful open source projects and interviews with key persons from these projects.

The results of the study indicate that despite the different natures of proprietary and open source software, the foundation of success is the same – the project needs a clear business model and a value proposition, something that seems to be lacking in Makumba's case. The differences in marketing work between proprietary and open source software are mostly centered to which marketing channels to use, as a result of the open source projects' lack of budget. Throughout the study, a number of potential marketing channels for Makumba are suggested.

Sammanfattning

Ett webbutvecklingsramverk är en grund för webbapplikationer, som innehåller den funktionalitet som ofta återfinns i webbapplikationer. Syftet med detta arbete är att utreda varför Makumba, ett webbutvecklingsramverk med öppen källkod som designats särskilt för att ha en låg inlärningströskel, inte har nått den förväntade mängden användare, och hur Makumba-communityt bör agera för att få marknaden att ta till sig projektet i större grad. Detta utreds genom studier av klassisk marknadsföringsteori tillsammans med fallstudier av ett antal lyckade öppen källkodsprojekt och intervjuer med nyckelpersoner från dessa projekt.

Studiens resultat visar att trots skillnaderna i användandet och utvecklingen av proprietär respektive öppen källkods-mjukvara, så är grunden till framgång densamma – projektet behöver en tydlig affärsmodell och ett värdeerbjudande, något som anses saknas i Makumbas fall. Skillnaderna i marknadsföringsarbete mellan proprietär och öppen källkods-mjukvara ligger främst i vilka marknadsföringskanaler man bör använda, som ett resultat av de finansiella begränsningarna hos öppen källkods-projekt. I studien föreslås ett antal potentiella marknadsföringskanaler för Makumba.

Table of Contents

1 Introduction

In this chapter the background to this thesis is discussed. The problem description for the research and its goals and purpose are also specified along with the delimitations that have been necessary and a short discussion on source criticism.

1.1 Background

Makumba is a framework for developing web applications, and in order to understand the concept, one first needs some basic knowledge about the World Wide Web and its development.

The World Wide Web was developed around 1990 in an attempt to enable for scientists to share large amounts of data and documents with each other. (Jazayeri, 2007) The web has since then evolved at a rapid pace, today having a great impact on everyone's everyday life by supporting document and workflow management in cooperative work, distributing all types of knowledge and media.

There are two major differences between web application development and traditional software development:

- The evolutionary nature of web applications increases requirements of frequent changes in content and functionality of the web application.
- Web application development timeframes are shorter due to the high volatility on the market and the evolving nature on web applications. (Murugesan, 2008, pp. 14-15)

These difficulties increases both supply and demand for tools facilitating the web development process, one of these tools being a number of different web application development frameworks. The idea of using a framework is based on that the foundations of many web applications are very similar, containing a user interface and managing a certain amount of data. The typical structure consists of:

- Data being displayed in the user interface, typically by HTML
- Client-side scripts running on the client, handling the interaction with the user
- Server-side scripts running on the server, handling the interaction with the database (Jazayeri, 2007)

Murray, Carrington & Strooper (2004) defines the framework concept as following:

> *"A Framework is a reusable design that requires software components to function."*

In other words, a web development framework is a foundation designed to help developers build web applications, providing core functionality appropriate for most web applications and thus saving time for web developers. Examples of functionality that can be provided by a web development framework are user session management, data persistence and templating systems (Jazayeri, 2007).

Many frameworks today use the Model-View-Controller, MVC, architecture. In MVC, every graphical user interface component is divided into three elements:

- *Model*, which is the application object, and stores all data for the object
- *View*, which is the screen presentation
- *Controller*, which handles all user input (Weisfeld, 2004)

Makumba is an open-source web development framework that was designed as part of a PhD thesis in 2003. Makumba was designed for the student organization Board of European Students of Technology (BEST), in an attempt to design a framework particularly appropriate for novice programmers. The choice of target users imply, among others, that the framework is easy to get started with, i.e. that it provides a short learning curve. (Bogdan, 2003) However, despite Makumba being easy to get started with (Öhrn, 2010), the framework has not reached any popularity outside the initial community.

1.2 Problem description

Why is Makumba having a hard time entering the market, and what actions can be taken by the Makumba community to simplify this entrance?

1.3 Goals and purpose

My goals are:

- To explain Makumba's lack of impact on the web developing framework market.
- To make relevant and concrete recommendations to the Makumba community as to what they should do management- and PR-wise to increase awareness and of Makumba.

The overall purpose of this thesis is to simplify the Makumba community's future promotion work.

1.4 Delimitations

In order to make this thesis feasible within the course framework, I have had to make some delimitations.

As will be discussed in chapter 3, I will not investigate the technological delimitations of Makumba, although these might exist. Examples of technological delimitations of Makumba would be requirements on support for certain techniques on the web server, which delimitates which web hosting servers that can be used.

Furthermore, I will not investigate the developer experience and/or organizational benefits achieved by using Makumba, mostly since it because of Makumba's low user basis is difficult to find enough data on this. This thesis is based on Makumba providing satisfactory developer experience and organizational benefits, i.e. that lack of awareness of the framework is the largest reason for Makumba not reaching its expected usage numbers.

1.5 Shortcomings

When carrying out empirical studies, in this case interviews, practical reasons affects which responses you get. For example, not all desired interviewees have been available due to practical reasons. However, the number of interviewees and the proliferation of projects that they represent are still satisfying.

During the case studies of other open source projects, only successful projects have been studied, i.e. projects that have gained a somewhat large user basis. As no unsuccessful projects have been studied, nothing can really be said about how to distinguish if a project will succeed or not. For example, a project might have carried out marketing in the same way that is suggested here, however yet not succeeding. What can be concluded by this study is limited to seemingly common characteristics of these successful projects; however these are no guarantee of success.

1.6 Target group

The target group for this thesis is mainly the Makumba community. However, anyone involved with any open source project could find it interesting

1.7 Source criticism

In order to answer the main questions in this paper, a comprehensive literature study has been carried out, as well as a number of interviews. The literature study is carried out with a vast number of sources of different kinds, such as books on the subject, articles published in scientific journals and websites representing the organizations and companies that are studied.

When possible, I have used sources that are reputable and influential on the subject, such as Philip Kotler's *Marketing Management* and articles published in leading journals on the subject or in databases such as acm.org, which is a leading provider of high-quality literature on the subject. Further, I have strived to use multiple sources to all theory, and particularly to the theory on which analysis and conclusions are based. When contradictions between the different sources are encountered, these are pointed out and discussed further.

Websites are used mainly for background information on the projects being studied. The objectiveness of this material cannot be guaranteed, as there is a risk for this information being promotional and subjective. However, this will not affect the results of the study as this material acts mainly as background information in order to better understand the study, rather than theory on which conclusions are being based.

2 Theory

In this chapter the necessary theory for the research questions is presented, covering open source in general, specific open source projects and marketing theory.

2.1 Definition of Open Source

Open Source Initiative, an organization dedicated to promoting the interests of open source software, states that open source doesn't just mean free access to the source code, but should also comply with a number of criteria, among these the following:

- *Free distribution*: The license shall not restrict any party from selling or giving away the software as a component of an aggregate software distribution containing programs from several different sources. The license shall not require a royalty or other fee for such sale.
- *Source Code:* The program must include source code, and must allow distribution in source code as well as compiled form.
- *Derived Works:* The license must allow modifications and derived works, and must allow them to be distributed under the same terms as the license of the original software. (The Open Source Definition)

As most open source software can be distributed freely, it is difficult to approximate the actual usage numbers of an open source software. It is possible to measure the number of downloads, but one download may be used by several users and on several computers. Furthermore, a download does not imply actual usage. The Debian Popularity Contest is the only existing solution to measure usage today, but it only measures the usage of the programs or packages installed on computers running the Debian operating system. (Bronner, 2007)

2.2 Makumba

Makumba is a web development framework with the following core ideas (Makumba.org):

- *Creating web-apps should be simple*, meaning that there should not be a need to have a deep technical background in order to create a simple web application.
- *Code should be sustainable*, meaning that the code needs to be readable, understandable and usable also by these who did not write it.
- *Everything is a query-fragment*
- *Breaking MVC is cool*, implying that different parts of the MVC pattern does not need to be strictly separated, but should be mixed in order to maximize effectiveness.

Makumba differs from most other web development frameworks through its view on the MVC pattern. As mentioned earlier, most frameworks today strictly follows MVC, but the Makumba community states that as many interactive web applications mostly are used to simply browse and update data from a database, there is no need for the strong model-view separation required by the MVC pattern. Makumba's interaction programming technology is instead centered on

database queries and does for example allow queries in views, with which repetition of data description is avoided and the separation between model and view is reduced. (Makumba.org)

One important aspect of Makumba is its quite narrow original target group; an amateur community with novice programmers, (Bogdan, 2003) which makes certain features more desirable and other features less relevant. For example, there should not be a need for having a deep technical background, why Makumba relies on simple languages such as HTML markup and a SQL-like query language called MQL, Makumba Query Language. (Makumba.org)

Makumba organizes a system in three parts:

- *The Makumba Data Definition*, which describes data structures and relations between them, in the form of simple lists of data fields with name and type.
- *The Makumba JSP Tag Library*, which is a tag library for displaying and editing the data which is described in Makumba Data Definitions and stored in database.
- *The business logic part*, written in Java, which restricts how the data can be edited. (Bogdan & Mayer, 2009)

2.3 Other common frameworks

2.3.1 ASP.NET: ASP.NET MVC

ASP.NET is based on the Active Server Pages technology and the Microsoft framework .NET. It is not free to use and requires an IIS, Internet Information Services, a server software released by Microsoft. (Karlsson, 2010) The ASP.NET MVC framework is strictly based on the Model-View-Controller pattern. Microsoft released the most recent version of the framework, ASP.NET MVC 3, in January 2011. (ASP.NET MVC)

2.3.2 PHP: CakePHP, Zend, Solar & Symfony

PHP is an open source development, with the purpose of creating a language that allows web developers to create dynamically generated pages in a quick and easy way. Its syntax draws upon C, Java and Perl, and the PHP script code is generally embedded into HTML. (www.php.net) (Vaswani, 2009, pp. 7-8) CakePhp, Zend, Solar & Symfony are all popular PHP frameworks.

2.3.3 Python: Django

Django is a Python-based open source framework developed by The World Company. It uses MVC, although not implemented in the classic way. The official documentation of Django separates the website into four main parts: Model layer, View Layer, Template layer and Forms. *Model layer* is identical to the MVC's definition of Model. *View layer* is a representation of the data being presented to the user, while *Template layer* handles the visual presentation. *Forms* is a code library handling forms and user input. There is no Controller as presented in the classical MVC definition, as this functionality is regarded as being built into the framework's underlying functionality. (Karsberg & Jönsson, 2010)

2.3.4 Ruby: Ruby on Rails

Ruby is a programming language that was fairly unknown until the release of the Ruby on Rails framework. Ruby on Rails is based on the following components:

- Action Controller, which represents the controller part of the MVC pattern.
- Action View, which represents the view part of the MVC pattern.
- Active Record, which represents the model part of the MVC pattern.
- Action Mailer, which is a framework for sending email in Ruby on Rails.
- Active Resource, which provides code for the usage of the Representational State Transfer (REST) principle, an IT architecture concept for machine to machine communications.
- Railities, which binds together the framework.
- Active Supports, which is a code library providing support functionality for the developer. (Karsberg & Jönsson, 2010)

Ruby on Rails is famous for being used by the micro blogging service Twitter. (Karsberg & Jönsson, 2010)

2.3.5 Perl: Catalyst

Perl is an open source programming language created by Larry Wall and first released in 1987. (Berman, 2009, p. 2) Catalyst is a Perl web application framework which was originally derived from another Perl framework called Maypole in 2004. Today, Catalyst is one of the most used Perl web application frameworks, and makes use of the MVC architecture. (Catalyst::Manual::Intro - search.cpan.org)

2.4 Some successful open source projects

2.4.1 Joomla

Joomla is an open source web content management system (WCMS). A content management system is a tool for handling large volumes of information, and a web content management system is a tool for creating and administrating web pages. The three dominating web content management systems on the market today are Wordpress, Joomla and Drupal. The Joomla project is a spin-off from a similar project called Mambo and was initially released in September 2005. (Gustafsson & Montgomerie-Neilson, 2010)

2.4.2 cURL

cURL is a command line tool for transferring data – getting or sending files – through URL syntax. (cURL and libcurl) Daniel Stenberg, the creator of cURL, estimates the downloads of cURL to approximately one million per year, but states that it is practically impossible to estimate as cURL also is distributed together with for example Linux. (Stenberg, 2011)

2.4.3 Rockbox

Rockbox is an open source replacement firmware for portable digital audio players, providing a range of extra features normally not included in the original firmware, for example battery time

indicators, screensavers, bookmarks in songs and extended resume features. It supports a number of different players, among others most models of Apple's iPod. (Rockbox - Open source jukebox firmware)

2.4.4 CanCan

CanCan is an authorization library for Ruby on Rails, providing functionality controlling which resources a given user is allowed to access. All permissions are defined in a single class, and does not need to be duplicated across contollers, views and/or database queries. The project has over 2000 watchers on Gihub. (Bates)

2.5 Market adoption of new technologies

In order to better understand the successes or failures of new product promotions, it is useful to study the market adoption of new technologies. The adoption of a new technology often follows an S-curve. (Fisher & Pry, 1971)

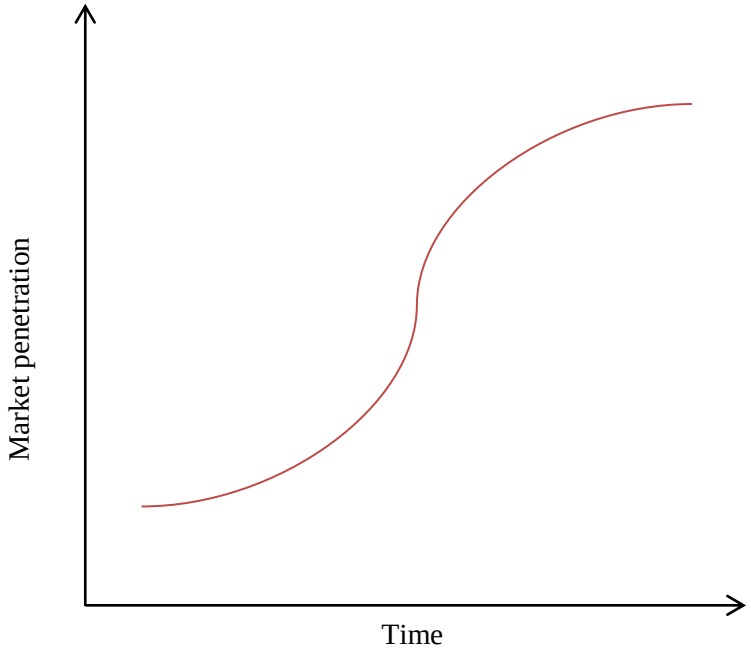

Figure 1: S-curve

The sudden and steep rise is an effect of herding behavior among humans –a rush to buy or sell a particular asset often prompts others to do the same, meaning that traders on an economic market influence one another directly. (Ball, 2004, pp. 215-217) The critical mass of a social system is reached when there is a momentum in the growth of the system, so that the system becomes self-sustaining and creates further growth.

This idea is closely related to the Tipping point concept presented by Malcom Gladwell, who is often criticized in academic circles. Criticism of Gladwell is mainly focused to Gladwell being a journalist rather than an academic, why his ideas should be used carefully in academic writing. However, with them being quite similar to Ball's idea of the critical mass as presented above, they are worth mentioning in this context. Gladwell describes the tipping point as *"the moment*

of critical mass, the threshold, the boiling point", and states that ideas, products and messages spread like viruses do. (Gladwell, 2000, pp. 7-12)

Gladwell further emphasizes the relevance of *The law of the few,* that implies that 80% of the work when spreading an idea is made by 20% of the participants. These 20% of the participants can be divided into three categories: connectors, mavens and salesmen.

- *Connectors* are the kind of people that seem to know everyone, and are known for introducing people who normally live and work in different social circles.
- *Mavens* accumulate knowledge and are known as "information specialists". We rely on these to provide us with information, and they often start word-by-mouth epidemics by sharing their knowledge.
- *Salesmen* are persuasive and charismatic people with good negotiation skills. (Gladwell, 2000)

This implies that when marketing a new product, in this case Makumba, the resources should be focused on reaching these three kinds of people.

Rogers (1983) also divided the adopters of a new technology into different categories:

- *Innovators*, the first ones to adopt an innovation. They have high risk tolerance, often due to financial resources, and can therefore adopt technologies with high risks of failure.
- *Early Adopters* are the second fastest category of individuals to adopt an innovation. They have an advanced education and have a high social status.
- *Early Majority* adopts the innovation after a varying degree of time. These people have above average social status and good contacts with early adopters.
- *Late Majority*, which will adopt an innovation later than average. They usually have a high level of skepticism against innovations.
- *Laggards*, typically elderly people focused on traditions.

2.6 Market adoption of open source software

Market adoption of open source software differs in some ways from market adoption of proprietary (closed source) software. The product likely is cheaper to use than its proprietary equivalent, which enables for smaller organization with less financial funds to use the software. Of course, also larger organizations seek to maximize economic viability, and are likely to prefer open source software because of its economic benefits; the price of a product is simply a key factor in the success of a market introduction. (Nagy, Yassin, & Bhattacherjee, 2010)

However, there are also barriers for the market adoption of open source software. Most of these can be deduced to the lack of economic funds of the organizations behind the products. Lack of awareness and/or knowledge about open source software is often referred to as the greatest barrier of adopting open source software. The lack of knowledge regards both the existence of the open source software, the technical knowledge needed to implement and use it, and the business knowledge needed to customize it.

Because of the developing organizations' lack of marketing or advertising budgets, there are less resources of reaching and informing the software's potential adopter base. (Nagy, Yassin,

& Bhattacherjee, 2010) Instead of using the established marketing avenues of commercial software distributors, open source software development organizations have to make use of word of mouth or other information sources to a greater extent. (Goode, 2005)

Furthermore, it has been discussed that it is a common perception that goods available for free, such as open source software, are likely to be of inferior quality than those that are paid for. (Nagy, Yassin, & Bhattacherjee, 2010) This needs to be considered when marketing open source software, as the product's quality might need to be extra emphasized.

2.7 Traditional marketing principles

In order to investigate the promotion of Makumba, it is useful to know some basic marketing principles. Most of the traditional marketing principles are based on the ideas by Philip Kotler, author of for example *Marketing Management* which is said to be the most widely used textbook in marketing education. Although the classic promotional ideas might not be applicable to this kind of product, the processes behind developing a marketing strategy is highly relevant.

Target markets and *segmentation* is a significant part of a market strategy according to Kotler; this since a marketer rarely can satisfy everyone in a market. A target market is a customer market segment which presents a particularly good business opportunity for the company. (Kotler, 2003, p. 9) The segmentations should be made based on the benefits that the customers in the segment seek from the product in question. In practice, however, the segmentation needs to be made based on observable characteristics, why we need to find a correlation between the benefits being sought and some observable characteristic, such as age or demographics. (Silk, 2006, pp. 86-87)

The firm is to develop a market offering for each target market, containing characteristics being positioned as central benefits in the minds of the prospective buyers in the segment. (Kotler, 2003, p. 9)

Kotler (2003, p.11) states that the offering will be successful if it delivers *value* to the target buyer. The definition of customer value varies somewhat throughout literature, but are broadly similar. According to Kotler (2003, p.11), value can be seen as being dependent on quality, service and price; increasing with service and quality and decreasing with price. Or, as follows:

$$Value = \frac{Benefits}{Costs} = \frac{Functional\ benefits + Emotional\ benefits}{Monetary\ costs + Time\ costs + Energy\ costs + Physic\ costs}$$

Other definitions of customer value throughout literature include:

> *"Value is the consumer's overall assessment of the utility of a product based on perceptions of what is received and what is given." (Zeithaml, 1988)*
>
> *"Buyers' perceptions of value represent a tradeoff between the quality or benefits they perceive in the product relative to the sacrifice the perceive by paying the price." (Monroe, 1990, p. 46)*
>
> *"By customer value, we mean the emotional bond established between a customer and a producer after the customer has used a salient product or service produced by that supplier and found the product to provide an added value." (Butz & Goodstein, 1996)*

Out of these definitions, we can establish that customer value indeed is related to the customer's perception of what is received, i.e. what the customer feels that he or she gains by using the product, related to what is given, i.e. what the customer pays for the product.

Kotler further makes use of the concept of the product life cycle model, in which the product goes through four phases:

1. *Introduction*, when the product is introduced in the market. Sales are experiencing slow growth and profits are nonexistent because of large costs associated with the product introduction.
2. *Growth*, a period of rapid market acceptance, increasing sales and improving profits.
3. *Maturity*, where sales slow down and profits stabilize or decline because of increased competition.
4. *Decline*, where sales and profits decline.

Each stage of the product life cycle calls for different marketing strategies. (Kotler, 2003, pp. 328-331) In this case, as Makumba clearly still is in the introduction stage, the marketing objective is to create product awareness and trial. A common strategy for this is to offer a basic product and advertise this to build product awareness among early adopters and dealers, in opposite to mass marketing which is useful in the growth phase. (Kotler, 2003, p. 340)

In conclusion, Kotler (2003, p. 630) specifies five steps in the development of an advertising program:

1. Set advertising objectives
2. Establish a budget
3. Choose the advertising message
4. Decide on which media to use
5. Evaluate the communication and sales effects of advertising.

2.8 Marketing 2.0 & Marketing 3.0

Buyers today have better possibilities of filtering the content being delivered to them, by using caller IDs to avoid unwanted phone calls, recording TV programs and fast-forwarding through commercials, or downloading movies and music (both illegal and legal alternatives) to escape cinema- and radio-commercials. This has led to response rates to traditional, pushing, market campaigns often being as low as 0.5%, which is ineffective from a Return of Investment (ROI) perspective. (Borgers, 2009, p. 1) Maximizing the ROI of a marketing effort is essential to all businesses, but especially to open source communities such as Makumba, as a result of their limited marketing budgets, as discussed in chapter 2.6.

The new types of marketing and promotion that followed Web 2.0 makes use of the network effects that are achieved through social media and are often called Marketing 2.0, as a reference to the Web 2.0 concept. This type of marketing is described as a two-way communication, in contrast to traditional promotion (Weinberg, 2009, pp. 3-4), and the consumer is seen as smart, in opposite to Marketing 1.0's view upon the market as mass buyers. (Kotler, Kartajaya, & Setiawan, 2010, pp. 3-6)

Furthermore, Kotler defines the latest marketing concept – Marketing 3.0 – as how a brand connects with the human spirit of its customers who desire that they assume their fair share of

social responsibility for issues such as environment, hunger, poverty, human rights, and so on. According to Kotler, this is the future of marketing (Kotler, Kartajaya, & Setiawan, 2010, pp. 3-6), however it is such a new concept that evidence of its effectiveness is not to be found.

Kotler et al (2010) concludes the differences between Marketing 1.0, Marketing 2.0 and Marketing 3.0 as follows:

	Marketing 1.0: Product-centric Marketing	*Marketing 2.0: Consumer-oriented Marketing*	*Marketing 3.0: Values-driven Marketing*
Objective	Sell products	Satisfy and retain the consumers	Make the world a better place
Enabling forces	Industrial revolution	Information technology	New wave technology
How companies see the market	Mass buyers with physical needs	Smarter consumer with mind and heart	Whole human with mind, heard and spirit
Key marketing concept	Product development	Differentiation	Values
Company marketing	Product specification	Corporate and product positioning	Corporate mission, vision and values
Value propositions	Functional	Functional and emotional	Functional, emotional and spiritual
Interaction with consumers	One-to-many transaction	One-to-one relationship	Many-to-many collaboration

Table 1: Summary of Marketing 1.0, 2.0 & 3.0 (Kotler, Kartajaya, & Setiawan, 2010, p. 6)

2.9 The business model concept

It is seemingly common for Open Source Software development to experience difficulties in generating value. (Dahlander & Magnusson, 2006) Chesbrough & Rosenbloom (2002) derive in part these difficulties from the lack of viable business models, and state that creating a viable business model is a critical and neglected dimension of creating value from technology. The ultimate role of the business model is said to be to ensure that the technological core of the innovation delivers value to the customer.

Chesbrough & Rosenbloom (2002) states that a business model should contain the following:

- A value proposition, i.e. the value that the users receives by using the framework
- A market segment, i.e. the users to whom the framework is useful
- A definition of the value chain structure within the firm that is required to create and distribute the value
- The cost structure and profit potential, given the value proposition and the value chain structure

- The organization's position within the value network, which links suppliers and customers to the organization
- A competitive strategy, describing how the organization will gain and hold advantage over rivals.

3 Method

In this chapter, the chosen work method is presented, both on a comprehensive level and in detail, covering selection of interviewees and interview questions. The method chosen will also be examined for validity and reliability.

3.1 Background

A common approach when considering market adoption of a new product is to model this as a funnel with a number of different steps. This can be applied to Makumba's case as following:

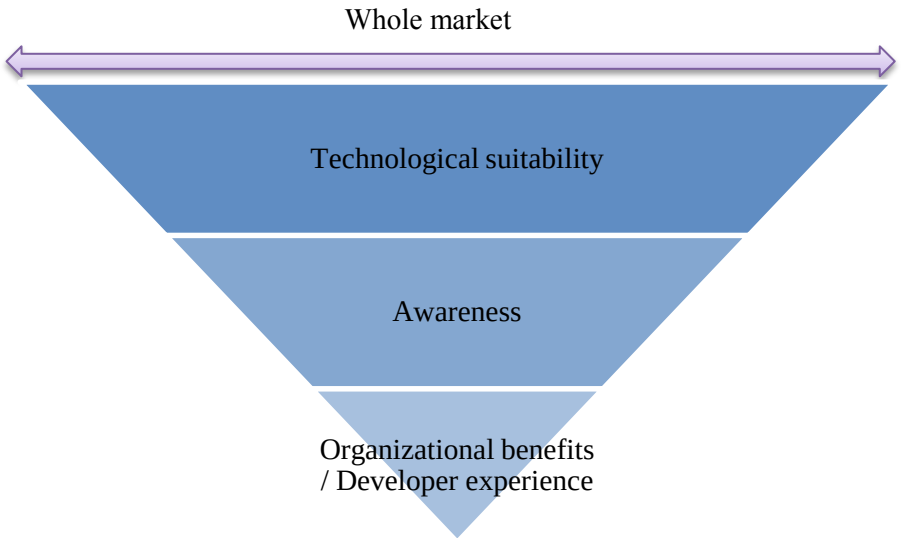

Figure 2: Model for market adoption of a new product

In words, the steps can be described as follows:

- *Technological suitability:* For which kinds of projects is the framework suitable? Are there any bottlenecks for the usage of the framework because of Makumba's technical solution? How can Makumba's technological possibilities be extended?
- *Awareness:* How many developers working on projects appropriate for Makumba are aware of the existence of the framework? How can the awareness of Makumba's existence among possible users be increased?
- *Organizational benefits / Developer experience:* How many of the developing organizations working on appropriate projects and aware of the framework would consider choosing Makumba as a development framework because it provides a pleasant developing environment for the organization? How can this developing environment be improved? This level can be viewed upon as a conscious choice from the developing organization.

In this thesis, the awareness part will be investigated, i.e.: *How can awareness of Makumba's existence be improved?*

3.2 Awareness and recognition of Makumba

As stated above, this thesis will examine how the awareness of Makumba's existence can be improved and how the Makumba community can act, management- and PR-wise, to promote the project. In order to gain understanding for how to promote open source projects successfully, a qualitative study is carried out where a number of successful (i.e. with a large user basis) open source projects are studied, and key persons within these projects are interviewed.

3.3 Interviews

3.3.1 Selection of interviewees

The selection of interviewees has been made based on:

- Which open source projects they are representing
- Practical reasons such as availability for interviews

The open source project that each interviewee represents, acts as a smaller case study as the factors behind this project's success is analyzed. As the purpose of the study is to find methods for the community to encourage the spreading of the software in question, I have chosen to study exemplary cases, i.e. cases showing the ideal, rather than representative cases, i.e. cases showing the typical. Therefore, the major selection criteria when choosing cases to study, has been whether or not the project has been successful, which I define as having reached an user basis larger than the critical mass.

Further, when carrying out interviews, all desired interviewees will due to practical reasons not be available, something that of course also has affected the selection of interviewees.

3.3.2 Interview questions / themes

The following are the main questions to which I have sought answers when carrying out the interviews:

- Introduction: The interviewee's background and projects.
- When a technology or product has reached a certain amount of users, it will partly spread itself through word by mouth (Critical mass theory, chapter 2.5.). But how can an open source community act to spread the first awareness of the project outside the initial community? How have the interviewee acted to achieve this initial awareness for his/her project?
- How can the marketing of an open source project be organized? How can the smaller user groups in each country contribute to the overall promotion of the project? Is it common to have a formalized marketing plan, or is marketing of successful open source projects carried our more ad hoc?
- Which concrete tools can be used for promoting open source projects? Which tools have been used by the interviewee?
- What spontaneous recommendations do the interviewees have for Makumba, upon taking a quick look at Makumba's website (www.makumba.org)?

3.4 Method validity and reliability

Validity and reliability in qualitative research has been widely discussed, and many propose different terms to be used in quality research, such as *credibility, transferability, dependability* and *confirmability*. However, it can be argued that by accepting the redefinition of the traditional terms, one acknowledges that a qualitative study is in fact unreliable and invalid. Instead, alternative ways of operationalizing the terms *validity* and *reliability* need to be found to suit a qualitative study. (Robson, 2002, p. 170) Robson proposes three main threats to the validity of a qualitative study:

- *Description* of what have been heard or viewed when carrying out the studies. Robson states that audio- or videotaping should be carried out when possible. As this has not been feasible in this study, extra attention have had to been paid to taking correct notes of the interviews and backing up conclusions with data from multiple sources.
- *Interpretation* of what have been observed during the studies. Robson states that the main threat is that of imposing a framework on what is happening. This can be avoided by not only describing one's conclusions, but also how they and their underlying interpretations were made.
- *Theory*, where the main threat is not considering alternative explanations of the phenomenon that is being observed. This can be countered by so called negative case analysis, which means actively seeking data that is not consistent with the proposed theory. (Robson, 2002, p. 171)

Further on, respondent and researcher bias need to be discussed. Respondent bias can appear in different shapes, such as withholding information when the researcher is seen as a threat, or the respondent trying to give the answers or impressions which they believe that the researcher wants. Researcher bias refers to for example assumptions and preconceptions owned by the researcher, which affects the study in terms of which interviewees are selected, which questions that are asked, or which data is selected for analysis. (Robson, 2002, p. 172)

Robson (2002) presents a number of strategies for dealing with respondent and researcher bias:

- *Prolonged involvement*, meaning that the researcher spends a long time, often a year, in the studied setting. This is not applicable to this study because of course framework timeframe.
- *Triangulation*, which involves using multiple sources. In this study, theory triangulation will be used, i.e. multiple theories with different perspectives will be used. Further on, multiple interviews will be carried out, with interviewees with different background, and data from all interviews will be used to analyze.
- *Peer debriefing / support* suggests that groups of students and/or researchers can contribute to avoiding researcher bias through debriefing periods. This has been done somewhat within the course framework.
- *Member checking*, i.e. returning to interviewees and presenting transcripts and interpretations that have been made. This will be done at the end of the thesis work.
- *Negative case analysis*, i.e. actively seeking data not consistent with the proposed theory. Negative case analysis is carried out when analyzing the data from the study.
- *Audit trail*, which suggests that a full record of all activities is kept. This has been done throughout the study.

Strategy	*Researcher bias*	*Respondent bias*
Prolonged involvement	Increases threat	Reduces threat
Triangulation	Reduces threat	Reduces threat
Peer debriefing / support	Reduces threat	No effect
Member checking	Reduces threat	Reduces threat
Negative case analysis	Reduces threat	No effect
Audit trail	Reduces threat	No effect

Table 2: Strategies for reducing researcher and respondent bias (Robson, 2002, pp. 172-176)

The objective of testing a study's reliability is to ensure that the study can be repeated with the same results. (Yin, 2003, p. 38) The concern is whether the tool used for observation, in this case the human, produces consistent results. (Robson, 2002, p. 176)

One of the most important aspects of reliability is to document the procedures followed in the earlier case. (Yin, 2003, p. 38) Ensuring reliability is achieved by not only being thorough and honest when carrying out research, but also through being able to show this to others, which is done by thorough documenting and keeping an audit trail. (Robson, 2002, p. 176)

4 Results and analysis

First, each interview will be presented and analyzed individually. Thereafter, the results from each interview will be compared and discussed in order to enable for drawing appropriate conclusions. All interviewees have approved of their names being mentioned in this thesis.

4.1 Interview with Lorenzo Garcia, Joomla

4.1.1 Garcia's background

Lorenzo Garcia is one of the founders and former president of The Swedish Joomla Organization (Svenska Joomlaföreningen), an organization dedicated to spreading knowledge about Joomla. Garcia's involvement in the Joomla community began with moderating the Swedish part of a Joomla forum, later becoming an editor of the Joomla extensions directory, which is the official gathering place for Joomla extensions (plugins). Garcia has furthermore been a board member of Open Source Matters, the legal organization behind the Joomla community, for several years.

4.1.2 Market introduction of Joomla

Since Joomla, as mentioned in chapter 2.4.1, is derived from Mambo, it received a vast amount of attention upon its first release and did not have to start from zero awareness, which is the case with Makumba. The Swedish Joomla Organization was founded by a core group of Swedish members from the Joomla forum, with the aim of spreading awareness of Joomla's existence in Sweden. At that time, promotion of Joomla internationally was quite wide, through press releases and similar channels, but nonexistent in Sweden.

Garcia instead exemplifies market introduction of an open source software with the case of Magento, a popular e-commerce solution launched by the American company Varien in 2008.

> *"During Varien's one year long development of Magento, it was frequently mentioned on the company blog on the company's website. This made Varien's existing customers the first who knew of Magento. These passed the news on further, as Magento appeared to be remarkably better than its open source competitors. Magento held a standard normally achieved only by expensive proprietary software and described this in a concrete way in the blog posts."*

The early blog posts of Magento can be found in the archive of Magento's website (http://www.magentocommerce.com/blog/).

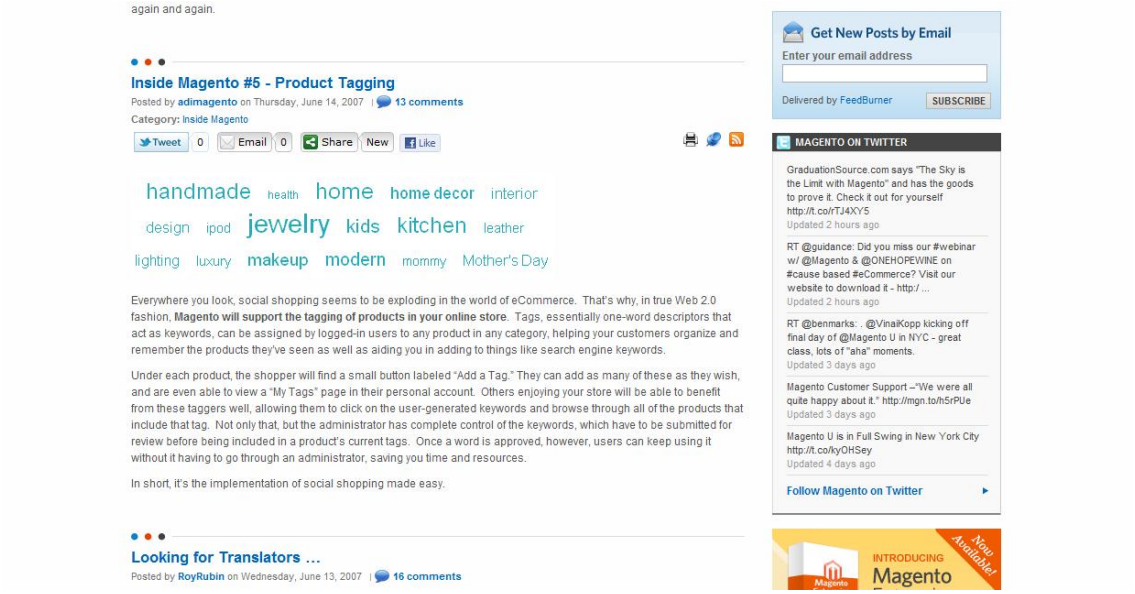

Figure 3: Example of a blog post from Magento, showing desirable features to be released with the software

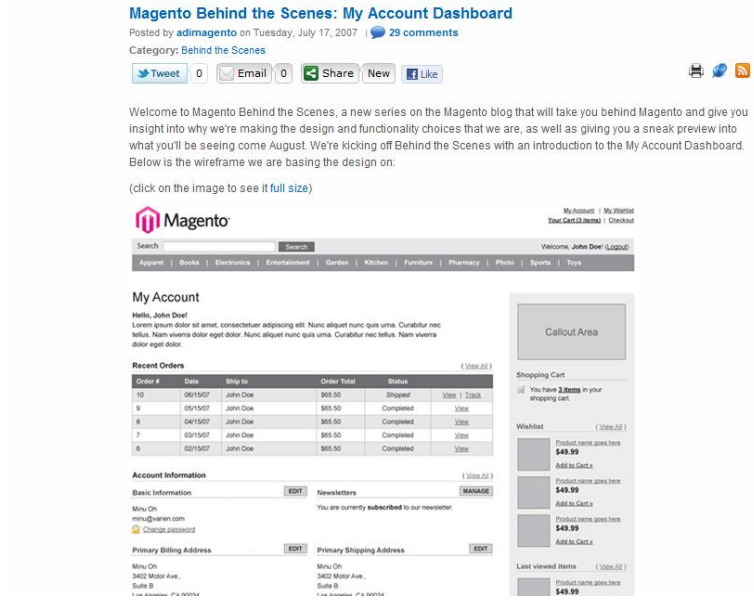

Figure 4: Example of a blog post from Magento, showing the administrative interface and its features

4.1.3 Organization of the marketing of Joomla

Garcia states that the coordination of the marketing work has been unsatisfying, particularly in Sweden. Internationally, there were originally two PR managers in the Joomla community, but their tasks were later taken over by Open Source Matters, who hired a professional PR bureau to handle the marketing of Joomla. During this period, traditional marketing methods were used to promote Joomla. Today marketing is handled within the Joomla community, although the PR manager positions are currently vacant.

The marketing carried out by The Swedish Joomla Organization has according to Garcia been fairly unorganized due to the nature of work on a voluntarily basis and the time constraints that

this entails. The marketing activities have been carried out through enthusiasts in channels available.

4.1.4 Tools for marketing Joomla

In a discussion of which concrete marketing activities that have been carried out by the Joomla community, Garcia mentions the following:

- Press releases through a free account at mynewsdesk.com.
- Contacts with and tips to relevant newspapers and websites such as idg.se.
- Events such as Joomla days and Joomla evenings with lectures and networking events. These events were promoted through Google Adverts, press releases, word by mouth and emails to potential users of Joomla. However, a majority of the participants of these events were already somewhat familiar with Joomla, resulting in networking of existing users being the primary function of these events rather than spreading awareness of Joomla's bare existence.

4.1.5 Garcia's recommendations for Makumba

Garcia states that a web development framework somewhat differs from a content management system in that a framework has fewer potential users than a content management system, as it is used by developers only, while a content management system can have users without developer skills, using the system for instance for creating a simple blog. Furthermore, the number of recognized web development frameworks is greater than the number of recognized content management systems. There are, as mentioned, three content management systems with a somewhat large user basis, and many more web development frameworks. Therefore, Garcia emphasizes the necessity of identifying Makumba's strengths and promoting these.

> *"The project must identify its strengths in order to know what to tell the potential users. Why should anyone choose this framework over another?"*

Further, Garcia accentuates the importance of extensive documentation and an active forum. However, in order to maintain an active forum where encountered problems can be discussed, a number of users are needed, but it is difficult to attract users without an already active forum, which becomes a vicious circle. The initial Joomla community worked actively with gaining an active forum:

> *"We were a number of key people who acted as support through the forum to the users who encountered problems. Some of these who were helped did in turn begin helping others, and after a while we could leave all the simpler matters to other users and gradually withdraw from the forum."*

> *"Attracting volunteers is a key to having a successful open source project. There is much work to be done apart from the actual coding, such as forum moderating."*

Garcia also recommends for the framework to try attracting certain key projects which can lead to more people using Makumba in order to achieve synergy effects. He once more exemplifies with Magento:

"Magento is based on the Zend framework. Therefore, the developers of Magento are familiar with the Zend framework and are likely to use the Zend framework also in other projects since they already know it well and want to achieve synergy effects."

4.2 Telephone interview with Daniel Stenberg, creator of cURL and RockBox

4.2.1 Stenberg's background

Stenberg has been running FOSS Stockholm, a community for Free and Open Source Software enthusiasts in Stockholm, together with Claes Jakobsson since December 2009. He is the creator of many open source projects, two of the most famous being cURL (chapter 2.4.2) and Rockbox (chapter 2.4.3).

4.2.2 Market introduction of cURL and RockBox

Stenberg states that cURL appears to have spread by itself already from the beginning, which he derives to the project being unique and filling a specific need. If a person requires the functionality provided by cURL and searches for a solution, cURL will be the only one he finds.

4.2.3 Organization of the marketing of cURL and RockBox

Stenberg acknowledges that he doesn't really do any active marketing for his projects, as he does not have any incentive to seek to maximize the number of users.

"It is of course fun to achieve many users, but 'many enough' is good enough. My occupation or other involvements are not dependent of this, financially or on other way."

4.2.4 Tools for marketing cURL and RockBox

As stated, Stenberg has not been carrying out any marketing activities in order to promote cURL and RockBox.

"The marketing have been carried out through the usual channels; uploading them on SourceForge and FreshMeat (open source software collector websites, ed.)."

4.2.5 Stenberg's recommendations for Makumba

"There are many more web development frameworks on the market than products similar to mine, a real jungle. Makumba has many more competitors than my projects, why my knowledge is not entirely applicable to web development frameworks."

Due to the vast number of competitors to Makumba, the importance of explaining which needs the product fulfills, and why the user should choose the project, increases. Stenberg is not convinced by the Makumba website:

"It is not apparent on the Makumba website what makes the framework special. Why should I choose Makumba?"

> *"I would consider choosing Makumba if they, preferably through a table or a graph, could show how they are technically superior to their competitors."*

Stenberg has two concrete suggestions for actions to be taken by the Makumba community in order to promote Makumba:

- Participating in open source conferences, industry sector conferences for potential customers, and other gatherings of potential future users of Makumba.
- Attracting certain high-profiled pioneering projects, working as a kind of quality certification. For example, the CMS Drupal early attracted The White House as their user. This kind of key projects give status to the project and can be used in the project's marketing.

Finally, Stenberg believes that the choice of which programming language to use is highly important.

> *"Java has traditionally had a hard time on the open source market, probably because Java itself wasn't originally open source. Java is more considered as an Enterprise solution, and few open source enthusiasts like Java."*

4.3 Interview with Claes Jakobsson, Stockholm Perl Mongers

4.3.1 Jakobsson's background

Claes Jakobsson is one of the founder and chairman of Stockholm Perl Mongers, a Perl User group for the programming language Perl, and is running FOSS Stockholm together with Daniel Stenberg. In the early 2000's, Jakobsson founded Stockholm Perl Mongers together with a colleague.

4.3.2 Market introduction of Perl

Jakobsson believes that Perl reached its critical mass around 1990, due to the appearance of the World Wide Web and Perl's suitability for coding CGI scripts.

> *"There were not many alternatives for writing CGI scripts at the time. Possibly you could use C or Ruby, however C is by many regarded as a more difficult programming language, and Ruby was not widely known at the time."*

Jakobsson further states that the initial awareness of Perl was spread through Usenet, a worldwide distributed Internet discussion system that was used before the introduction of the World Wide Web.

4.3.3 Organization of the marketing of Perl

There is someone who is responsible for coordinating marketing in the international Perl community, however the marketing is not planned long term but is carried out more ad hoc.

> *"The marketing responsible is mostly responsible for making sure that there are representatives at the big conferences, and that PR material is being provided."*

The main activities of Stockholm Perl Mongers have been starting up, arranging and participating in the yearly Nordic Perl Mongers conferences with approximately 100 participants. The first workshop was held in 2003 with participants from Sweden and Denmark.

> *"The Nordic Perl Mongers conference is a forum for networking and spreading ideas within the Nordic countries. Sweden is too small!"*

Jakobsson further states that the Swedish Perl community is fairly inactive and is currently not carrying out other activities than the yearly Nordic conference. However, according to Jakobsson, other Perl Mongers groups such as London Perl Mongers are larger and have a higher frequency of events. There are also Perl conferences for larger areas such as Europe and North America. Jakobsson further emphasizes the importance of the social part of this kind of gatherings:

> *"My primary incentives to participate in these events are to network and socialize with like-minded."*

As Perl is a programming language, in the same way as Java, which Makumba is based on, it is more interesting to discuss a few Perl frameworks or libraries, as these are of more similar nature as Makumba. Jakobsson explains that all modules and frameworks in Perl are distributed through a Perl Network called CPAN (www.cpan.org). According to Jakobsson, it is common for a Perl enthusiast to check out the new releases at CPAN, especially the ones created by renowned people within the community, and the new releases are often initially discussed at the different IRC channels.

4.3.4 Jakobsson's recommendations for Makumba

> *"Twitter would probably be the most important marketing channel for Makumba."*

Nonetheless, in order for someone to read your tweets, i.e. in order for being able to use Twitter as an effective marketing channel, you need to have somewhat of a follower basis already. Therefore, it might be easier for a person already known in the target circuit to spread awareness of a new project through mentioning it on Twitter.

> *"It could be an idea to get news of Makumba posted on Slashdot (an American website with many news on among others open source, ed.), but the Makumba community needs to be prepared to defend its project as posting on Slashdot often lead to wide discussions and heavy critique in the comments system."*

Similarly to Stenberg, Jakobsson believes that Makumba will have an extra hard time as it's a Java framework.

> *"First of all, Java doesn't work well on Apache. Second, there are already a vast amount of Java frameworks, such as Spring, that are developed by big companies with greater resources."*

Therefore, Makumba needs very unique selling points; it needs to be very clear to the potential user when for example visiting Makumba's website why he or she should use Makumba instead of Spring and the other big Java frameworks.

> *"When looking at Makumba's website, Makumba appears to be just another framework in the crowd. Why is Makumba a better choice than any other Java framework?"*
>
> *"A possible strength for Makumba is its stated simplicity. Everyone wants a simple and hands-on solution."*

Makumba's selling points needs to be adverted at Makumba's website, and further, Jakobsson believes that there must me more life on the website. He exemplifies with the website of the Perl framework Mojolicious (http://www.mojolicious.org/):

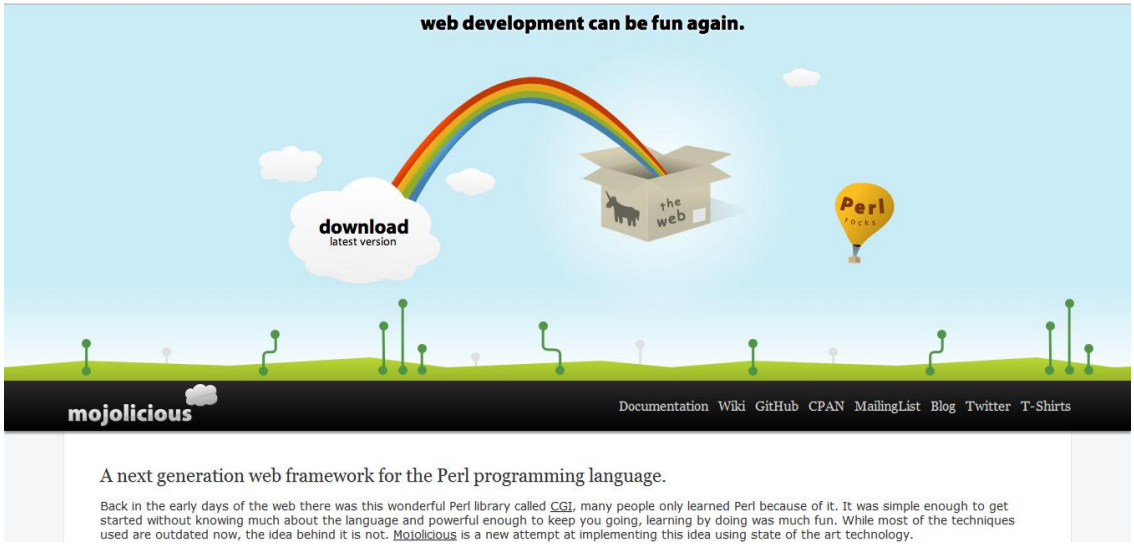

Figure 5: Front page of Mojolicious's website

Jakobsson states that Mojolicious's website has the following that Makumba's website is missing:

- A mailinglist
- A frequently updated blog with news on what's happening in the framework development, what features will be included on the upcoming releases, when will these releases be, etc.
- A twitter
- Prop sales
- A innovative and funky design

Jakobsson emphasizes the importance of a practical usage connection with concrete usage examples when promoting an open source software. Jakobsson suggests that the Makumba community for instance makes a video of how simple the framework is to use, and how fast you can solve certain problems with it.

Another issue with Java is that there is no central source for information, where releases of new modules and frameworks such as Makumba, can be posted. As mentioned earlier, all Perl modules and frameworks are distributed through CPAN.

> *"If you are looking for a Perl module with a specific functionality, you go to CPAN and search for it."*

> *"Sourceforge (where the Makumba source code currently is posted, ed.), is not the best distribution place today. Makumba should use Github instead."*

Finally, Jakobsson states that much of the information on new frameworks and modules is spread on conferences and through the networks developed at these events, and he therefore suggests that the Makumba community participates in conferences such as JavaOne and JavaWorld.

4.4 Email response from Ryan Bates, creator of CanCan and Ruby on Rails activist

4.4.1 Bates's background

Ryan Bates is the creator of the Ruby on Rails authorization library CanCan (chapter 2.4.4). Through email, I asked Bates to share his thoughts on the questions of this thesis.

4.4.2 Market introduction of CanCan

Bates states that an open source project needs to be promoted when released in order to gain users.

> *"Simply uploading it is not enough, it's like creating a website and expecting people to visit it just because it's online."*

4.4.3 Tools for marketing CanCan

Bates recommends the following tools for promoting an open source project:

- Twitter
- Blog posts, showing how to use it
- Talking about the projects at conferences
- Ask external podcasts and news sites to mention the project

Further on, Bates emphasizes the need for a demand of the product:

> *"This all assumes you have a quality project that fits a specific need better than anything else out there. Be certain it has good documentation and is well tested."*

4.4.4 Bates's recommendations for Makumba.

Bates states that the Makumba website is nice but, doesn't appear to be alive.

> *"You have a nice website with a quick start guide, so that's good. […]It's good that the project has a blog, but it doesn't look good when the last entry is 8 months ago. It makes me wonder if the project is still alive and supported."*

Like Stenberg, Bates prefers GitHub to SourceForge:

> *"I personally like GitHub much more than Source Forge and I will sometimes choose a project solely on where it's hosted."*

4.5 Email response from David Heinemeier Hansson, creator of Ruby on Rails

4.5.1 Heinemeier Hansson's background

David Heinemeier Hansson is the creator of Ruby on Rails, and I contacted Heinemeier Hansson to ask questions about the market introduction of Ruby on Rails, as it is by many regarded as similar to Makumba in many aspects.

4.5.2 Market introduction of Ruby on Rails

Heinemeier Hansson's tactic for spreading initial awareness of Ruby on Rails was similar to that of Varien and Magento (chapter 4.1.2):

> *"So I started talking about Rails on the Ruby mailing lists about 6 months before it was officially released. By the time it was released, there was a lot of initial energy to try it within the community."*

4.5.3 Organization of the marketing of Ruby on Rails

Similar to several of the other organizations being studied, there doesn't seem to be a formalized marketing organization:

> *"There's no organization of it, per se."*

4.5.4 Tools for marketing Ruby on Rails

It needs to be clear from the start in which way the project is superior to its competitors.

> *"How is Rails better? I picked two competing platforms to benchmark against: PHP and Java. In a series of blog posts, I showed how much simpler Ruby on Rails was to use. I criticized both platforms loudly for their failings."*

> *"I made it ultra easy to understand what Ruby on Rails was about with the famous 15-minute screencast. It got a lot of people interested in trying it because I was showing how easy it was to use, not just talking about it. Show don't tell."*

4.6 Analysis

4.6.1 Market introductions of successful open source projects

The open source projects that have been studied in this thesis have had varying conditions for their respective market introductions. Joomla, for instance, was already somewhat known among its potential users because of its derivation from Mambo, which has not been the case with some of the other projects. Further, it appears that a project developed by a person already known in the circuits of the target user group, automatically becomes more interesting. For example, Jakobsson states that it for Perl users is common to check out CPAN for new releases by certain key people.

As for the projects that were started entirely from scratch and without a certain key person as attraction, many of these projects, such as Magento and Ruby on Rails, had in common that they were promoted and discussed on blogs and podcasts long before their initial release, which created an initial energy and curiosity for the project. However, this is of course not possible to recreate retrospectively.

An interesting contradiction that is encountered is the question of whether or not the project actually needs to be promoted. Bates states that simply uploading a project isn't enough for it to gain users, but Stenberg states that he has not done more than just that for his projects. He means that users have found his projects as they were the only ones solving that particular issue. Similarly, Jakobsson states that Perl spread through being the only usable alternative for writing CGI scripts. Seemingly, if the demand for the basic functionality being provided is large enough, awareness of the project will spread itself from the beginning as users will find the project when searching Google or software collector sites for solutions to the problems they are encountering. This assumes that the project in question really is the only existing solution to a particular problem in the existing conditions. Often, this is not the case, but the project might be the *best* solution to a particular problem in certain specific conditions, although the project is not the *only* possible solution.

This idea correlates well with the marketing 2.0 concept, where differentiation is the key marketing concept and the consumer is being seen as smart, making rational choices based on the information available. (Kotler, Kartajaya, & Setiawan, 2010, p. 6)

4.6.2 Organization of marketing work done by open source organizations

None of the studied organizations appear to have formalized organizations of their marketing work. The general opinion seem to be that the success of the project is more dependent on the demand for the functionality of the project than having a formalized marketing organization, even if some, like Stenberg, states that it of course serves to have a thorough marketing plan (at least, it won't do any harm).

The difficulties of formalizing the marketing organization are discussed. The main problems lie in the nature of volunteer work – people do this in their free time and will therefore not give it maximum priority. Furthermore, the involved want devote their time to carrying out concrete activities, from which they can see actual results, rather than devoting their time to writing a formalized marketing plan. It is shown throughout the study that it is more the rule than the exception that marketing of open source project is carried out rather "ad hoc", in what can be described as a "quick and dirty way".

4.6.3 Thoughts on Makumba

All interviewees believe that it is difficult to penetrate the market for web development frameworks because of the vast number of frameworks already on the market, especially Java frameworks. Although the different frameworks each have different advantages, their basic function is similar, if not identical, why a certain web development framework will never be the only solution to a particular problem, as discussed in chapter 4.6.1. Instead, because of their different characteristics, a particular framework will be the best solution to a particular problem in certain conditions. This implies that the demand for the functionality being provided itself will not be enough for attracting users, as was the case with cURL, RockBox and to some extent

Perl. The Makumba community will need to work harder to advertise their advantages and unique selling points.

None of the interviewees have heard of Makumba prior to being approached in this study, but several of them visited www.makumba.org prior to the interview to find out more about the framework. The general impression of Makumba gained from this visit is that Makumba is "just another framework in the crowd" and that Makumba's advantages compared to other frameworks are not apparent. Makumba's market offering and unique selling points need to be adverted on the website in a more concrete and hands-on way.

As stated in chapter 2.7, Kotler (2003) argued that a market offering will be successful if it delivers value to the target buyer. Furthermore, Chesbrough & Rosenbloom (2002) stated that creating a viable business model is of great importance in order to create value to the customer. It appears that the Makumba community needs to create or reconsider its business model in order to identify the value being delivered to the customer compared to its competitors, i.e. its unique selling points. First after finding Makumba's unique selling points, can one advertise these and, as a result, stand out from the crowd. All interviewees emphasize the need for concrete examples on why Makumba is better than another framework, for example through a graph, table or images.

Another issue with Makumba's website is that it does not appear to be updated frequently enough. Bates states that this makes him wonder if the project is still alive and supported. The blog should be updated frequently with previews of new releases, new features to be included, and practical usage examples. Furthermore, all interviewees emphasize their desire for hands-on usage examples, similar to Heinemeier Hansson's 15 minute Ruby on Rails screencast.

Garcia and Bates accentuate the importance of having extensive documentation for the project, which is not regarded to be entirely the case with Makumba. The lack of documentation is acknowledged by Bogdan & Mayer (2009), who argue that Makumba's lack of documentation contributes to peer learning by forcing new members of the IT group to resort to their colleagues for help. However, the lack of documentation reduces Makumba's attractiveness as a solution in settings without earlier usage, where the knowledge and experience of the framework is reduced or non-existent. In other words, the lack of documentation increases the barrier for new organizations to adopt the framework. As discussed in chapter 2.6, lack of awareness and/or knowledge about open source software is considered to be the greatest barrier of adopting open source software. (Nagy, Yassin, & Bhattacherjee, 2010) Furthermore, other studies have found that documentation is indeed an important tool for communication (Forward & Lethbridge, 2002).

Finally, several of the interviewees suggest attracting key projects or users, which can bring several benefits:

- If another open source project is using Makumba, the developers of that project will then continue to use the framework also in their other project, as was the case with Magento and the Zend framework.
- If a certain high-profiled project is using Makumba, this can work as a quality certification and be used in the project's marking, as was the case with Drupal and The White House.

As for key users, Gladwell's (2000) *The law of the few*, implying that that 80% of the work when spreading an idea is made by 20% of the participants, which was discussed in chapter 2.5, implies that the resources should be focused on reaching the key persons involved in these 20%. Also, initial marketing should, to the extent that it's possible, be targeted at the presumed early adopters of Makumba. (Rogers, 1983)

5 Discussion

Here, the found application of traditional marketing to open source projects will be discussed in a speculative manner, followed by a short discussion on the creation of a viable business model for Makumba.

5.1 Applying traditional marketing methods to open source projects

It has been discussed that open source projects might not able to use traditional marketing methods as a consequence of their lack of financial resources. Nevertheless, this study has shown that the basic principles of traditional marketing still stand – the project will be successful if it manages to create value for the customer. Although none of the studied projects have had formalized marketing plans defining their value proposition, all interviewees have mentioned the importance of having a value proposition, although in more casual terms. Likely the formal term is not used simply because of the interviewees' lack of a theoretical background within the marketing discipline. Instead, it is referred to as being more of "common sense" that you need to know what your project is offering the potential user.

Seemingly, traditional marketing and marketing of open source projects are more alike than expected, as it all goes down to the simple question of delivering value to the customer and showing the potential customer that value. However, the nature of volunteer work leads to labor in the form of time and commitment from volunteers being the limiting factors. This, perhaps in combination with lack of a theoretical background of the volunteers, leads to marketing within open source communities today being carried out in a "quick and dirty way". It can be discussed whether this is optimal use of the resources, but it appears that minimizing the formalization works as long as the projects are not too large, which could lead to misunderstandings and communication issues within the organization.

Apart from the desire to minimalize work not directly related to promotion activities that exists in non-benefit organizations, the largest difference from traditional marketing is which marketing channels that are appropriate to use. This is limited by the lack of budget of the open source projects, but the nature of the projects also affects what channels that are suitable. In Makumba's case, a television ad would likely not be very effective, as the majority of the viewers would be outside of Makumba's target market. An article in a respected online computer magazine however, is likely to reach more potential users, as potential users of Makumba, web developers, because of their interests are likely to read these kind of magazines, thus making this a far more efficient – and likely cheaper – advertising channel.

5.2 A business model for Makumba

It has been stated that the Makumba community needs a viable business model. The business model concept was discussed in chapter 2.9, and I settled for the definition made by Chesbrough and Rosenbloom, where a business model should contain:

- A value proposition
- A market segment
- A definition of the value chain structure
- The cost structure and profit potential
- The organization's position within the value network
- A competitive strategy

While all of these are important parts of a business model, some can be regarded as being more directly linked to the marketing work in particular:

- Makumba's value proposition, as the community needs to know *what* to tell the potential users.
- Makumba's market segment, as the community needs to know *who* to tell about Makumba.
- A competitive strategy, as the community needs to position Makumba against its competitors when creating the value proposition.

The technical factors will play the major role when comparing Makumba against other frameworks in the work with the three points above. Thus, in order to discuss these further, one needs to discuss the technical aspects of Makumba which haven't been covered in this thesis.

6 Conclusions

In this chapter, the results of the study will be concluded.

The Makumba community needs to create, reconsider or specify its business model in order to identify the value being delivered to the customer compared to its competitors. This knowledge about Makumba's unique selling points is necessary for standing out in the large crowd of web application frameworks on the market.

Makumba's unique selling points should be advertised on www.makumba.org and through other information channels. Emphasis should be on concrete evidence on how Makumba is superior to its competitors, for example through performance graphs and tables, and hands-on usage examples, preferably short screencasts.

A number of concrete actions to be taken by the Makumba community have been encountered. The following can be considered to be the most significant:

- www.makumba.org needs to feel more alive, so that potential users can see that the project is still supported. The blog should be frequently updated with news, including screenshots and videos, on what's happening within the development of the framework.
- A twitter account
- A mailing list
- Press releases (for example through mynewsdesk.com)
- Contacts with and tips to relevant newspapers and websites such as idg.se, as well as external podcasts
- Organizing events and workshops with lectures and networking events
- Considering using GitHub instead of or as a complement to SourceForge
- Mentioning the projects at conferences
- Attracting key projects and key users
- Creating an extensive documentation
- Developing an active forum, where the users can ask for advice and discuss solutions for various issues

7 Future work

Here, potential topics for future work related to the findings of this thesis will be suggested.

As concluded, the most important further development of this thesis is to create a viable business model for Makumba, which requires closer examinations and comparisons of Makumba's and its competitors' technical aspects.

Furthermore, I initially modeled the market's adoption of Makumba as a funnel that narrowed down with three aspects: technological suitability, awareness and developer experience. In this thesis, we have only investigated the awareness part. Although the three aspects are more or less intertwined, the Makumba community would benefit from investigating also the other aspects further, i.e. the following topics:

- *Technological suitability:* For which kinds of projects is the framework suitable? Are there any bottlenecks for the usage of the framework that arise to the Makumba's technical solution? How can Makumba's technological possibilities be extended?
- *Organizational benefits / Developer experience:* How many of the developing organizations working on appropriate projects and aware of the framework would consider choosing Makumba as a development framework because it provides a pleasant developing environment for the organization, and how can this number be increased?

8 Bibliography

ASP.NET MVC. (n.d.). Retrieved 03 21, 2011, from The Official ASP.NET Website: http://www.asp.net/mvc

Ball, P. (2004). *Critical Mass.* New York: Farrar, Straus and Giroux.

Bates, R. (n.d.). *ryanb/cancan - GitHub*. Retrieved 05 22, 2011, from GitHub: https://github.com/ryanb/cancan

Berman, J. J. (2009). *Perl: The Programming Language.* Jones's and Bartlett's Ltd.

Bogdan, C. (2003). *IT Design for Amateur Communities.* Stockholm: Royal Institute of Technology.

Bogdan, C., & Mayer, R. (2009). Makumba: the Role of the Technology for the Sustainability of Amateur Programming Practice and Community. *4t International Conference on Communities and Technologies* (pp. 205-214). Pennsylvania: University College.

Borgers, B. (2009). *Marketing 2.0: Bridging the Gab between Seller and Buyer through Social Media Marketing.* Wheatmark.

Bronner, M. (2007). *Measuring the Usage of Open Source Software.* Department of Applied Information Technology. Göteborg, Sweden: IT University of Göteborg.

Butz, H. E., & Goodstein, L. D. (1996). Measuring Customer Value: Gaining the Strategic Advantage. *Organizational Dynamics, 24*, 63-77.

Catalyst::Manual::Intro - search.cpan.org. (n.d.). Retrieved 05 05, 2011, from CPAN: http://search.cpan.org/~bobtfish/Catalyst-Manual-5.8007/lib/Catalyst/Manual/Intro.pod

Chesbrough, H., & Rosenbloom, R. S. (2002). *The Role of the Business Model in Capturing Value from Innovation: Evidence from Xerox Corporation's Technology Spinoff Companies.* Boston: Harvard Business School.

cURL and libcurl. (n.d.). Retrieved 04 15, 2011, from cURL and libcurl: http://curl.haxx.se/

Dahlander, L., & Magnusson, M. G. (2006). Business Models and Community Relationships of Open Source Software Firms. In J. Bitzer, & P. J. Schröder, *The Economics of Open Source Software Development* (pp. 111-130). Elsevier.

Fisher, J. C., & Pry, R. H. (1971). A Simple Substitution Model of Technological Change. *Technological Forecasting & Social Change, 3*, 75-88.

Gladwell, M. (2000). *The Tipping Point: How little things can make a big difference.* Little Brown.

Goode, S. (2005). Something for nothing: Management rejection of open source software in Austraila's top firms. *Information & Management, 42*, 669-681.

Gustafsson, T., & Montgomerie-Neilson, G. (2010). *Content Management Systems: Utvärdering av mjukvara med öppen källkod.* Stockholm: Royal Institute of Technology.

Jazayeri, M. (2007). Some Trends in Web Application Development.

Karlsson, T. (2010). *Konstruktion av en mindre webbplats i PHP-ramverket CodeIgniter.* Linköping: Linköpings Universitet.

Karsberg, M., & Jönsson, P.-J. (2010). *Ruby on Rails eller Django?: En jämförelse av två ramverk för webbutveckling.* Stockholm: Royal Institute of Technology.

Kotler, P. (2003). *Marketing Management.* Prentice Hall.

Kotler, P., Kartajaya, H., & Setiawan, I. (2010). *Marketing 3.0: From Products to the Customer to the Human Spirit.* John Wiley & Sons.

Makumba.org. (n.d.). Retrieved 02 14, 2011, from Makumba.org: http://www.makumba.org

Monroe, K. B. (1990). *Pricing: Making Profitable Decisions.* McGraw-Hill.

Murray, L., Carrington, D., & Strooper, P. (2004). An approach to specifying software frameworks. *Proceedings of the 27th Australasian conference on Computer Science.* Darlinghurst: Australian Computer Society, Inc.

Murugesan, S. (2008). Web Application Development: Challenges and the Role of Web Engineering. In G. Rossi, D. Schwabe, O. Pastor, & L. Olsina, *Web Engineering: Modeling and Implementing Web Applications.* Springer.

Nagy, D., Yassin, A. J., & Bhattacherjee, A. (2010, March). Organizational Adoption of Open Source Software: Barriers and Remedies. *Communications of the ACM, 53*(3), 148-151.

Öhrn, S. (2010). *Makumba and Ruby on Rails: Comparing two web development frameworks.* Stockholm: Royal Institute of Technology.

Robson, C. (2002). *Real world research: a resource for social scientists and practitioners.* Blackwell Publishing.

Rockbox - Open source jukebox firmware. (n.d.). Retrieved 04 05, 2011, from http://www.rockbox.org/

Rogers, E. M. (1983). *Diffusion of Innovations.* New York: Free Press.

Silk, A. J. (2006). *What is Marketing?* Harvard Business School Publishing Corporation.

Stenberg, D. (2011, 04 15). (C. Nilsson, Interviewer)

The Open Source Definition. (n.d.). Retrieved 03 26, 2010, from Open Source Initiative: http://opensource.org/docs/osd

Vaswani, V. (2009). *PHP: A beginners guide.* McGraw-Hill.

Weinberg, T. (2009). *The New Community Rules: Marketing on the Social Web* (First edition ed.). O'Reilly.

Weisfeld, M. A. (2004). *The object-oriented thought process.* Sams Publishing.

www.php.net. (n.d.). Retrieved 02 10, 2011, from Php.net: www.php.net

Yin, R. K. (2003). *Case Study Research: Design and Methods* (3 ed., Vol. 5). Sage Publications, Inc.

Zeithaml, V. A. (1988). Consumer perceptions of price, quality, and value: A means-end model and synthesis of evidence. *Journal of Marketing*, 2-22.