



Universitatea Tehnică a Moldovei

Elaborarea unui proiect cu sursă deschisă de succes

Developing a successful open source project

Student:

Dragoș Străinu

Conducător:

Silviu Gîncu
lect.univ.,
dr.

Chișinău 2020

Rezumat

Acest memoriu explicativ a fost elaborat pentru cercetarea și documentarea unui ghid pentru elaborarea proiectelor cu sursă deschisă de succes prezentat de Dragoș Străinu ca proiect de masterat la Universitatea Tehnică din Moldova. Scopul lucrării presupune investigarea elementelor importante care sporesc și mențin popularitatea unor proiecte cu sursă deschisă. Determinarea componentelor principale cât și cele secundare. Determinarea etapelor importante dar și acțiunilor care trebuie să fie întreprinse de către creatorii/mentinătorii proiectelor.

Teza este compusă din 4 capitole care au fost scrise în timpul dezvoltării proiectului și care sunt:

- **Domain analysis:** în acest capitol a fost analizat domeniul ”proiecte cu sursă deschisă”, influența acestui domeniu la întreaga industrie a tehnologiilor informaționale. Importanța lui pe viitor, și sfaturi pentru cei care de abea încep contribuțiile la proiecte cu sursă deschisă;
- **Important advices:** acest capitol are o serie de sfaturi pe care creatorii cât și mentinătorii proiectelor cu sursă deschisă trebuie să le considere pentru ca proiectele să aibă succes;
- **Components:** descrierea componentele principale și secundare care trebuie să le aibă un proiect (ca documentație, canale de comunicare, giduri, conferințe). Definirea instrumentelor alternative;
- **Stages of open source projects:** acest capitol identifică etapele de viață a proiectelor cu sursă deschisă și pașii care trebuie să fie urmați la fiecare etapă pentru a crește popularitatea și mis, care la următoare etapă.

Abstract

This explanatory memo was elaborated for the research and documentation of a guide for the elaboration of successful open source projects presented by Dragoş Străinu as a master's project at the Technical University of Moldova. The aim of the paper is to investigate the important elements that increase and maintain the popularity of open source projects. Determination of both main and secondary components. Determining the important stages but also the actions that must be undertaken by the creators / maintainers of the projects.

The thesis consists of 4 chapters that were written during the development of the project and which are:

- **Domain analysis:** in this chapter was analyzed the field “open source projects”, influential, this field to the entire information technology industry Jonah. Its important in the future, and advice for those who are just starting to contribute to open source projects;
- **Important advices:** This chapter has a number of tips that both creators and maintainers of open source projects need to consider for projects to be successful;
- **Components:** description of the main and secondary components that a project must have (such as documentation, communication channels, guides, conferences). Defining alternative tools;
- **Stages of open source projects:** this chapter identifies the life stages of open source projects and the steps to be followed at each stage to increase popularity and movement to the next stage.

Table of contents

INTRODUCTION	8
1 DOMAIN ANALYSIS	9
1.1 History	.9
1.2 Open-source software vs Free software vs Freeware	.10
1.3 Should you launch your own open source project?	.11
1.3.1 Setting goals	.12
1.3.2 Contributing to other projects	.12
2 IMPORTANT ADVICES	13
2.1 Who cares about open source	.13
2.2 Solve a real problem	.13
2.3 Put an accent on quality	.13
2.4 Choosing a license	.14
2.5 Choosing the right name	.15
3 COMPONENTS	16
3.1 Primary Components	.16
3.1.1 GIT	.16
3.1.2 Github	.17
3.2 Secondary Components	.17
3.2.1 Documentation	.17
3.2.2 Contribution guides	.18
3.2.3 Community development	.21
3.2.4 Forums	.22
3.2.5 Presentations, conferences	.23
3.2.6 Tutorial and articles	.23
3.2.7 Logo and branding	.24
4 STAGES OF OPEN SOURCE PROJECTS	26
4.1 Concept	.26
4.2 Bootstrap	.27
4.3 Early Development	.27
4.4 Early Adoption	.27
4.5 Development	.27
4.6 Adoption	.27
4.7 Maturity	.28
4.8 Mainstream	.28
CONCLUSIONS	29
REFERENCES	30

INTRODUCTION

As early as the 1960s, the source code of software products was delivered to the consumer to allow them to fix problems and add new features. In particular, software products created and used by universities imposing the principles of knowledge sharing. Companies also delivered the source code because sometimes customers ran the product on other devices and it was necessary to change the code to run smoothly on customers' devices.

Over the years the cost of producing software products has increased considerably relative to hardware, manufacturers were facing solutions already delivered with hardware. The software was not licensed at the time and was distributed with the source code in the public domain. Then, when software licenses had already appeared, there were cases when the software products were distributed without paying the license fee. To increase revenue, companies stopped delivering source code and delivered only executable code that was compiled from source code. In response to this move, licenses such as the GNU General Public License and copyleft were created to allow people to use free open source software. However, in the 20th century, most companies kept the source code private in order to sell software consulting and improvement services.

With the advent of versioning systems like GIT and code distribution platforms like Github, it has reduced the barrier to participation in free open-source products. Potential contributors no longer have to search for source code for products on independent sites. In the 21st century already many products are open-source and most private solutions are just a thin layer above the multitude of open-source solutions.

With the popularization of the open-source movement, hundreds of products and solutions appear daily on the open-source market. Github, the most popular open-source project distributor, has hundreds of thousands of repositories. Some projects are very successful with communities of tens of thousands of users. Others do not enjoy such success for various reasons. It is obvious that an open-source project is successful if it is useful to users. But utility is not the only decisive factor in the success of a project. At certain stages in the life of an open-source project, some steps are needed to promote this project and strengthen the solutions market. Some open-source projects, even if they were the first to solve problems and propose strong solutions, lost to other alternatives because they did not make the most of the means to attract and retain consumers.

References

10. S. Initiative, "Open source initiative." <https://opensource.org/> (accessed on 27-November-2020).
- 2D. Pavlutin, "How to make your open source project successful." <https://dmitripavlutin.com/how-to-make-your-open-source-project-successful/> (accessed on 27-November-2020).
- 3GitHub, "Starting an open source project." <https://opensource.guide/starting-a-project/> (accessed on 27-November-2020).
- 4Linux, "Linux kernel." <https://www.kernel.org/> (accessed on 27-November-2020). 5"Bitkeeper." <https://www.bitkeeper.org> (accessed on 27-November-2020).
- 6R. Hat, "Understanding the open source software life cycle." <https://www.redhat.com/cms/managed-files/cp-open-source-software-life-cycle-brief-f23003pr-202004-en.pdf> (accessed on 27-November-2020).
- 7O'Reilly, "Leadership in the open source life cycle." <https://www.oreilly.com/library/view/open-source-for/0596101198/ch01s08.html> (accessed on 27-November-2020).
- 8LosTechies, "Lifecycle of an open source project." <https://lostechies.com/chadmyers/2009/05/28/lifecycle-of-an-open-source-project/> (accessed on 27-November-2020).
- 9S. Lahtinen, "Open source project life cycle and survival." [https://samsai.eu/pdf/Open 20source % 20life%20cycle %20and %20survival.pdf](https://samsai.eu/pdf/Open%20source%20life%20cycle%20and%20survival.pdf) (accessed on 27-November-2020).