DuD • Datenschutz und Datensicherheit 3 | 2009

Bernhard Esslinger

CrypTool – an open source project in practice

Lessons learned from a successful open source project

Only a few open source projects can look back on a development period of more than ten years. CrypTool is one such project and has run through the entire gamut – from an internal company development, to freeware, to open source. How did CrypTool become a successful open source project?

**Background**

In recent years, open source software has become established as an increasingly feasible alternative to commercial products or internal software development projects in a corporate context.

Both the number of projects and the number of code lines from open source projects are experiencing exponential growth [Deshpande, Riehle 2008]. Well-known examples, such as the Apache web server, the Mozilla Firefox internet browser or the Linux operating system demonstrate that complex, high-quality IT systems and applications are being developed and continuously improved by numerous developers as open source projects.

The advantages of open source projects are obvious: most of the developers work on a voluntary basis and the results are often of better quality thanks to the high level of motivation and commitment of those involved. Additionally, users are normally continuously involved in the development process.

Thus, particularly in the light of increasingly costly internal development projects, it often seems a very attractive option to develop and maintain individual applications as open source projects.

However, despite all the advantages mentioned, we should not disregard the fact that the vast majority of open source projects are largely neglected after just a few months. With the exception of major, well-known projects, many open source projects are either inactive or rather unsuccessful [Chengalur-Smith, Sidorova 2003].

This is because distributed open source projects place special requirements on participants with regard to motivation. Communication and coordination efforts required for open source projects are generally much higher than for comparable internal development projects.

What are the key factors for a successful open source project? To answer this question, the experience of the CrypTool open source project is considered in detail below. Criteria are identified which are regarded as critical to the success of the CrypTool project.

**The CrypTool project**

The objective of the CrypTool project is to develop the CrypTool e-learning application for cryptography and cryptanalysis. With the aid of CrypTool, many cryptographic procedures can be applied and analysed to provide the user with a practical demonstration of basic and advanced concepts of cryptology.

CrypTool’s development began in 1998 as part of an IT security initiative for staff training purposes at Deutsche Bank. CrypTool has been available as freeware since 2000 and since 2002 it has been distributed on the German Federal Office for Information Security’s (BSI) citizen CD “Sicher ins Internet” (On the internet – with security) [Esslinger, DuD 2002]. CrypTool can also be downloaded from the BSI site for internet security [BSI for citizens]. It was only in 2003 that it became an international open source project [Suedmeyer, Esslinger 2008].

Thus, CrypTool did not start out as an open source project: It went from being a company internal development project to having freeware status, and then became an open source project [CrypToolPres 2008]. This transition made it possible to observe very closely not only the differences but also the advantages and disadvantages in the development process.
After three years as freeware, in 2003 CrypTool eventually became an open source project. While companies and universities were already using CrypTool as freeware, the direct involvement and integration of voluntary developers only became possible with the open source status. The open source approach improved the cooperation with universities, in particular, and numerous add-ons were developed for CrypTool in the course of seminars and theses. The University of Darmstadt took over the hosting of the project. In addition, the universities supplied concrete proposals for improving the user interfaces (e.g. experimental user interfaces from the University of Koblenz and the Aachen University of Applied Sciences). CrypTool 1.4.x still includes the commercial libraries as closed-sources (with the exception of the header files which are by necessity open). Even though this is not in the spirit of the open source concept, the approach is a pragmatic one: The functional scope can be extended professionally and the providers of the closed-sources offer the files free of charge for use in conjunction with CrypTool.

There are currently two parallel projects for the further development of CrypTool:

- JCrypTool (JCT) was originally also licensed under GPL 2. For Milestone 2, the team decided to changeover to the Eclipse Public License (EPL) version 1.0. The main reason for this was to achieve the highest possible level of conformity with the Eclipse standards. All the official Eclipse projects (i.e. all the projects run by the Eclipse Foundation) must be published under the EPL – JCrypTool wants to keep this option open for the future. The majority of JCT plug-ins are also published under this licence. The EPL is recognized by both the Open Source Initiative (OSI) and the Free Software Foundation (FSF).

- Problems of open source licences
  - The differences between the licences are not easy to grasp for non-lawyers. From the developers’ point of view, there are too many different types [Open source licences].
  - For CT1, the GNU General Public License Version 2 (GPL 2), the most widespread and well-known licence at the time, was chosen.
  - CT2 uses the Apache 2 licence. This licence was chosen for the following reasons: a disadvantage of the GPL 2 is that the executable file (*.exe) can only be linked to libraries which are also licensed under GPL. Strictly speaking, this would have meant that all plug-ins would also have had to use the GPL. Under the Apache licence, it is possible to load plug-ins with other licences. Moreover, while everybody may adapt and use the CrypTool code in any way, the name “CrypTool” can only be used for the original version. Thus, if someone were to develop their own application based on the current CrypTool code, the new programme’s name may not bear any resemblance to “CrypTool”. The Apache 2 licence has been recognized by the Free Software Foundation (FSF) as a free software licence and is compatible with the GPL version 3.

CrypTool was initially the result of a company internal development project. The application was used by many employees, but further development was not required for use within the company. However, the commitment shown by the developers led to the decision to continue CrypTool, albeit without any financial assistance. Consequently, CrypTool was made available to the public as freeware.

The transition to freeware initially proved difficult because of numerous legal problems, for example, export restrictions for various cryptographic algorithms. Moreover, CrypTool contained a series of libraries owned by commercial providers (e.g. Secude and Cryptovision) as well as a number of patented algorithms (e.g. IDEA encryption). Initially, the quality of the source code also posed a hurdle to the transition to open source.

However, once the transition to freeware had been achieved, the experience was extremely positive and made the additional effort worthwhile. A large number of users sent in suggestions for improvements or reported application errors, which enabled the developers to continuously improve CrypTool. The publication also benefited the company which, in addition to the favourable publicity, caught the attention of more IT experts as CrypTool. The “original sponsor”.

Further development only in

- JCT 1.0 (Java, Eclipse RCP, SWT)
- CT 2.0 (C#, .NET, WPF)

Hosting: SourceForge, University of Darmstadt, University of Duisburg-Essen
Localisation: University of Madrid, University of Warsaw
Distribution: global; more than 50,000 downloads annually
Current project participants: 45
Release version: CT 1.4.30 (C++)
Further development only in
- JCT 1.0 (Java, Eclipse RCP, SWT)
- CT 2.0 (C#, .NET, WPF)
The two projects came about primarily because a technology changeover was required: CrypTool 1.4.x was based on the outdated GUI library Microsoft Foundation Classes (MFC) and no longer met the project requirements. The decision was therefore taken to change over to modern development environments, which were more suitable for the distributed development. Additionally, the changeover meant that it was possible to align the project to current IT training in companies and universities and thus attract more developers. The free availability of software development tools was important to ensure that no additional barrier was created for the developers.

A survey was also conducted via the CrypTool website in which 40% of those questioned chose C# and .NET and another 40% chose Java and Eclipse. This decision coincides with recent labour-market studies (January 2009), which indicate that experts in Java and C#/.NET are particularly highly sought after [source Heise News 2009].

While the division into two projects may have initially appeared problematic, the parallel development has produced many good ideas, which were often taken up by the other line of development. The permanent CrypTool architects are active in both projects, which ensures that conceptual improvements are exchanged between the two development lines. Most of the voluntary developers had a clear preference for just one of the development environments so that selecting a single platform would probably have meant that a large number of developers would have been lost.

From an economic point of view, it would have been difficult to justify two parallel projects in a business context; however, as open source projects they tend to be more profitable and make it possible to experiment with different environments and approaches.

Success factors

Why is the CrypTool project still active? What are key factors for a successful open source project?

General motivation for the CrypTool project

A key question concerning open source is the motivation of individuals to participate voluntarily in an open source project and to sustain their level of participation over time. There have already been numerous empirical studies in this area – an overview can be found, for example, in Stoll 2006.

The main source of motivation identified in the studies is usually the acquisition and development of specialist knowledge and skills. As far as participation in the CrypTool project is concerned, an interest in cryptography and cryptanalysis is also clearly at the forefront. Cryptology covers a variety of processes and methods – ranging from classic encryption (e.g. Caesar) to modern algorithms and methods such as RSA, AES or lattice reduction. Cryptology is a basis technology for data protection and security and thus a key component for a basic understanding of the broader area of IT security.

For the developers, the challenge lies in correctly implementing what are in some cases very complex algorithms and also in visualising the methods suitably in order to make them comprehensible to laypeople. Implementing the algorithms allows the developers to expand their technical knowledge while, at the same time, the large number of experts working on the project provide support, in particular with software engineering problems.

A further incentive for new developers is the freedom to select a modern software development environment. Some developers use the CrypTool project as a way of familiarising themselves with one of the two supported development environments (in this case, Microsoft Visual Studio or Eclipse). Here the intrinsic motivation, namely enjoying programming, is accompanied by a desire to acquire expertise and expand professional knowledge in modern distributed development systems (and qualify for associated career opportunities).
Many developers come across the CrypTool project in the course of seminars or theses. In addition to the practical implementation experience, another attraction is the integration of the results into an existing environment and the advantage offered by the fact that the results of the academic work will live on within the project and will be developed further.

The additional effort of becoming familiar with an existing project is rewarded with valuable feedback from experts, which helps improve the quality of the individual’s own work. A certain ability to take constructive criticism is often required. However, the academic work produced has often been of far higher quality and greater relevance so that these undergraduates/postgraduate students have found it very easy to get a good job afterwards.

**Degree of freedom versus rules**

A key aspect of the participation of voluntary developers in an open source project is freedom regarding the implementation of functionalities, both in technical terms and with regard to the timing. Nevertheless, rules are essential to ensure that this freedom does not lead to creative chaos but instead, after a certain amount of time, produces worthwhile results.

Further guidance is given by the project Wikis which include documents listing and prioritizing the desired/missing functions.

In technological terms, CrypTool is based on an open framework approach that allows the developers to develop plug-ins which can be added to the overall application with relative ease. Plug-ins provide the functions of an encryption process, for example. This approach makes it possible to develop specific implementations of cryptographic procedures independently and then integrate them into the overall project at a later date. This enables developers to work on the project after a relatively short period of familiarization. The experience of implementing a plug-in also allows the developers to gain an understanding of the overall architecture and enables them to qualify for participation in the development of the actual core application. Another advantage of the plug-in approach is that the implementation is planned independently of the core application development. This means that the release planning essentially focuses on the core application and the plug-ins available at the time of the release are integrated into the overall package. If a user has installed an existing core application, he can download new plug-ins at any time with the aid of the built-in update function.

**Project organization and project management**

Many open source projects have a similar organizational structure, which resembles a typical software development project.

The project manager also has a key role to play in the CrypTool project. As the success of the project depends to a very large extent on the motivation of the voluntary developers, the project manager’s primary task is to sustainably motivate those working on the project. In particular, this means responding to any problems or questions reliably and promptly so that no misunderstandings occur. For new members the project initially appears to be a very anonymous construct. Therefore questions, especially from new members, are answered quickly, which in turn demands a high level of availability from the project management.

In the early stages of the open source project, attempts were made to decentralize tasks and to forward enquiries directly to the entire team. However, this did not prove a very positive experience as this highly decentralized communication approach often led to late or even conflicting responses. With the more centralised communication approach of the CrypTool project, it became clear that the project manager needs to have...
a good overview of the modules’ status and the developers’ skills. As the first point of contact the project manager also has to be able to provide information on general technical questions.

To distribute the tasks more effectively, CrypTool is currently considering the idea of establishing sub-project managers for individual categories (e.g. classic or modern encryption methods). This should not only make communication and decision-making more efficient, it should also help to boost motivation and should serve as recognition of past performance.

A key factor in the success of the CrypTool project is the interplay of different talents. For example, in addition to the architects and developers, there are also numerous mathematical, cryptological, software-engineering and web-design experts working on the project. It is the project manager’s task to ensure effective cooperation between the project staff and also encourage former developers to retain an interest in the project over the long term to enable efficient troubleshooting. Many former CrypTool developers still offer their services as advisors to the project.

In many projects, a particular challenge is posed by the more unpopular jobs, the main one being the documentation of results. We have seen that a clear allocation of tasks upfront can help. Consequently, every developer is required to document her/his code adequately.

In general, the requirements concerning the organization and the management of an open source project like CrypTool are comparable to those of a normal major software development project. Prior project management experience has also proved to be a very important aspect for open source projects, while motivation and communication play an even more central role in the open source context.

Agile software

- Successfully distributed open source projects usually have an agile software development, without this being explicitly forced through development tools: The entire software development process is repeatedly run through in short cycles so that new versions of executable software can be made available promptly, and these can be used to check whether the requirements are fulfilled.

Internal and external communication

Effective and efficient internal and external project communication is critical to the success of the CrypTool project.

In the JCrypTool project, the main form of internal communication is e-mail, while the CrypTool2 project has had very positive experience with an open Skype chat channel. As the project size remains very manageable, standard discussion forums are almost never used.

In addition, there is also a Wiki and a project website, however, these are mainly for external communication to provide users and developers with an outline of the project and the team, and to make the documentation available centrally (e.g. FAQs or rules or guidelines for the development) [JCrypTool][CrypTool 2].

Project meetings have proved very effective. However, it is generally very difficult to find a date which suits the scattered team. For this reason, the meetings are used primarily to agree the release plan.
ning and long-term architectural deci-

sions. The meetings are also very impor-
tant for promoting community spirit as

the individuals otherwise only communi-
cate with each other electronically.

As in the case of many other open
source projects, English has become estab-
lished as the project language for internal
communication even though the majority
of developers are from Germany.

For long-term, open source develop-
ment projects like CrypTool, new project
staff continually has to be found and, as
far as possible, retained on a long-term ba-
sis. Successful external communication is
key to this.

The project is attempting to increase
awareness of CrypTool not only through
numerous publications and presentations,
but also by participating in mathematical
and computer science seminars and work-
shops for interested and gifted pupils,
thereby attracting more users and, above all,
more developers. As a rule, the success
of the external communication can be
measured by the number of hits on the web-
site or the number of downloads. For ex-
ample, the download figures tend to increase
significantly following an announcement
on the Heise news ticker – which is very
popular in the German IT community –
that a new version of CrypTool is available,
and the team receives a great deal of feed-
back. In addition to having an appealing
website, it is particularly important that
queries are answered promptly. This is im-
portant not only for the developers, but also
for the users, and creates positive mouth-
to-mouth publicity.

The CrypTool project has also been suc-
cessful in a number of competitions. For
example, CrypTool was awarded the Euro-

pean Information Security Award and the
NRW IT security prize, among others. In
addition, CrypTool was chosen as a “se-
lected landmark” as part of the nationwide
initiative “Germany - Land of Ideas 2008”.

Prizes and publications are, however,
only of limited use for recruiting new de-
velopers as the subject of cryptology is
very specific and, as a rule, attracts only a
small group of users.

More important in this respect is that
CrypTool is used in the course of teaching
at a large number of universities as docu-
mented, for example, in the annual report
of Royal Holloway, University of London
[RoyalHolloway 2008]. At universities,
new developers are mainly recruited
through seminars or in connection with
theses. In this context, cryptographic al-
gorithms and methods can be implement-
ed as plug-ins, with the developers able to
build on an existing infrastructure or ap-
lication framework, enabling them to
achieve usable results more quickly and
focus on the actual algorithms during the
implementation. The CrypTool project has
found it very helpful to contact universi-
ties or the relevant professors directly and
propose cooperations. This was how the
Polish version of CrypTool by the Univer-
sity of Warsaw and the Spanish version by
the University of Madrid came about, al-
though the universities approached the
CrypTool project themselves with the idea
of offering their students a better learning
environment in their native language.

Technical infrastructure

As in any open source project, a technical
infrastructure, which ensures an efficient
distributed development, is required.

While SourceForge is perhaps the most
well-known provider of open source host-

ing, numerous free alternatives exist – e.g.
from Google (Google Code) or Microsoft
(CodePlex). These providers offer various
functionalities for managing projects
(open source hosting), including both
simple source-code hosting as well as ser-
vices for web hosting, Wiki, tracker, mail-
ings lists or forums.

The new CrypTool projects are hosted by
SourceForge [J CrypTool] and the Uni-

versity of Duisburg-Essen [CrypTool2].
Subversion is used as the version control
system. The system configuration is open
but subject to certain restrictions (anony-
mous read and download access for all,
write access only for specified accounts).
This allows users to easily accept modifi-
cations, while ensuring that the existing
code basis is not at risk through acciden-
tal changes.

Outlook

The transition from a company-internal
development project to an open source
project has proved very advantageous for
CrypTool. Thanks to the international
availability, numerous co-operations have
been established with universities in Ger-
many and abroad and new developers have
been found for the project.

However, handing over the develop-
ment to the open source community is not
in itself a guarantee of success. For the
long-term success of an open source pro-
ject, in addition to maintaining the gen-
eral motivation level, it is important to
create a suitable environment in which the
project can thrive and develop. This aspect
should not be underestimated and re-
quires not only commitment but also a
certain level of patience and degree of te-

nacity. As in almost any other project, it
ultimately comes down to the people in-
volved.

Bibliography


[Chengalur-Smith, Sidorova 2003] – “Success and Survival of Open Source Projects: A Popula-
tion Ecology Perspective,” in Proceedings of the 24th ICIS

Note 1: of the approximately 180,000 (as per Au-
gust 2008) projects in SourceForge (SF), 95% are “almost dead” (a rather harsh term for
projects which fulfill one or more of the follow-
ing conditions: the software is no longer
maintained (stays-as-it-is); creation and last
modification within the space of one year; no
further updates since the end of 2007; func-
tionality in the scope of a practice measure;
development status: “1 – planning” or “2 –

pre-alpha”; no response from the specified e-
mail address).

Note 2: unfortunately, lots of projects were started
without checking whether something similar
already existed, without any reference to such
existing projects or any attempt to merge
with them in order to provide the users with
added value.

Note 3: on 22 January 2009, JCT was ranked at 992
out of the 180,000 SF projects, indicating a rel-
atively high level of activity. However, this
fluctuates sharply.

Note 4: under a search for “Cryptography”, SF
found precisely 65 projects on 22 January
2009.

[CrypToolPres 2008] – a presentation of the entire
CrypTool project and its possibilities is availa-
ble at http://www.cryptool.org

vs.uni-due.de/

[Deshpande, Riehle 2008] – “The Total Growth of
Open Source”, in Proceedings of the Fourth Con-
ference on Open Source Systems (OSS
2008), Springer Verlag, 2008, pages 197-209.

Einstieg in klassische und moderne Kryptog-
raphie. Neue Version - fundierte Awareness in

Deutsch und Englisch,” published in “DuD
Datenschutz und Datensicherheit,” volume
10/2002, pages 615-618, https://www.cryp-

Java-und-C-Net-langfristig-gefragt--/meldung/121615


