Development of Safety-Critical Software Systems Using Open Source Software - A Systematic Map

Sardar Muhammad Sulaman, Alma Oručević-Alagić,
Markus Borg, Krzysztof Wnuk, Martin Höst
Department of Computer Science, Lund University, Sweden.
Email: {sardar, alma, markus.borg, krzysztof.wnuk, martin.host}@cs.lth.se

Jose Luis de la Vara
Simula Research Laboratory,
Norway.
Email: jdelavara@simula.no

***Abstract*—The popularity of Open Source Software (OSS) has increased the interest in using it in safety critical applications. The aim of this study is to review research carried out on usage of open source code in development of safety-critical software and systems. We conducted a systematic mapping study through searches in library databases and manual identification of articles from open source conferences. We have identified 22 studies about using open source software, mainly in automotive, aerospace, medical and nuclear domains. Moreover, only a few studies present complete safety systems that are released as OSS in full and that the most commonly used OSS functionalities are operating systems, imaging, control and data management. Finally most of the integrated OSS have mature code bases and a commit history of more than five years.**

***Index Terms*—Open source, safety critical, mapping study**

I. Introduction

Open Source Software (OSS) denotes normally software that "is distributed under terms that comply with the Open Source Definition" [1]. That is, the software is distributed according to a license that agrees with the Open Source definition. This means in practice that OSS is developed free of charge through a community-driven development process, and provided to users at no cost, but under certain usage and distribution conditions (license). OSS has been available for many years, and covers many different domains.

There are also many commercial companies that show an interest in participating in open source development and using open source components in their products [2]. Research is for example available on why and how companies participate in development communities (e.g. [3] where survey based on a sample of 300 projects from SourceForge is presented), what business models are common (e.g. [4] with a timeline for what has happened with a "hybrid" software system), and how open source components are chosen (e.g. [5] where it is investigated how integration issues are handled by companies using open source components). This means, as can be expected, that OSS is used in industrial commercial projects with all the industrial requirements. However, in for example the review presented in [2] it was not clear to what extent OSS is used in safety critical software systems.

"Safety critical systems are systems where it is essential that system operation is always safe" [6]. Examples of the systems are medical equipments, e.g. [7], control and monitoring systems in aircrafts, and automotive systems. Certification is an important part in development of safety-critical systems. That is, development is carried out in a regulated environment where the development complies with formal standards, regulations, directives and guidance [8]. Standards are available from a number of different bodies, e.g., ISO, FDA, etc., and certification bodies are available to check compliance with the standards. Even if development is certified there are a number of challenges in safety critical development listed by Leveson [9], such as the fact that technology is changing fast, the ability to learn from experience is limited, increasing complexity and coupling, more complex relationships between humans and automation, changing regulatory and public view of safety, etc.

The above mentioned issues and the regulated environment where safety-critical software is developed means that it is interesting to study and asses the extent of OSS usage within the safety-critical system development. Using OSS in the development of safety-critical systems may help to better learn from experiences, cope with changing technology and tackle the increasing complexity. There are some difficulties in using OSS in this type of systems, for example that certification may be more difficult if OSS is used, there is less control over OSS software, and that it is harder to investigate the correctness of the software. However, one can also argue that many OSS products are of high quality and "industry-standard" in some fields, and that the high usage results in high quality.

To summarize, there is a need to investigate what research has been conducted on the usage of OSS for safety-critical systems. The recent literature surveys about safety-critical systems focus on provision of evidence for safety certification [10] and on management of evidence for compliance with safety standards [11]. Focusing on adaptation of OSS in safety-critical software will give an understanding of to what extent this type of code is included in safety-critical systems, as well as an identification of further research areas.

The outline of this paper is as follows. In Section II the research methodology is presented. Results are presented and discussed in Section IV. In Section V conclusions are presented.

II. Methodology

The research presented in this study is carried out as a systematic mapping study conducted based on the guidelines

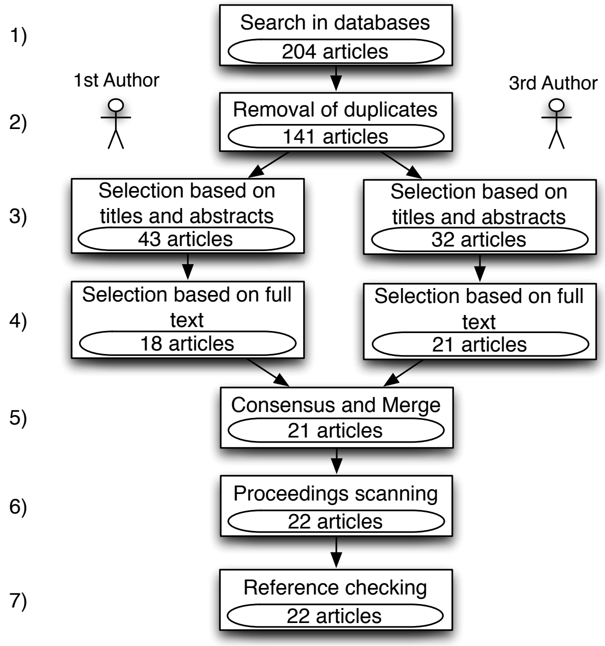

Fig. 1. Overview of the selection of primary studies.

presented by Kitchenham and Charters [12]. Figure 1 shows an overview of the steps involved in the selection of primary studies, which are also described below.

1) Applying the search query to the selected databases. This resulted in 204 articles.
2) Removal of 63 duplicate articles by using the *JabRef*[1] reference manager complemented by manual search, resulting in 141 remaining articles.
3) The first and third authors independently applied the inclusion and exclusion criteria based on titles, keywords and abstracts. This resulted in 43 and 32 articles.
4) Final selection of articles based on careful reading of full text, which resulted in 18 articles for the first author and 21 articles for the third author.
5) Conclusion of a final set of 21 articles.
6) Manual check of all OSS conference proceedings (from 2005 to 2012). This resulted in one additional article.
7) Manual check of all references of the 22 articles. This did not result in any additional article.

A. Research questions

The main objective of this study is to investigate what (empirical) research has been carried out on the usage of OSS in the development of safety-critical software systems. With usage of open source we mean both development of open source software and the inclusion of open source software in developed products. However, we do not include research on usage of open source tools, such as Eclipse or Libre Office in the research.

The main objective has been broken down in the following research questions:

RQ1: In which safety-critical systems' domains have the OSS systems been used?

RQ2: What types of OSS functionality have been included in safety-critical systems?

RQ3: What are the characteristics of the communities developing the OSS identified in RQ2?

B. Search strategy

We applied two main approaches in order to identify relevant articles: a broad search in academic databases, and a manual scanning of all articles published in the proceedings of the Conferences on Open Source Systems together with a search of articles based on the references of the identified articles.

Both INSPEC and the COMPENDEX databases provide a good coverage of the area, and include articles from all major conferences, journals, and publishers (e.g. IEEE, ACM, Springer, and IEE). We believe that these two databases give a good coverage of articles in "computer science" and "electrical engineering and electronics". We constructed the following search string to conduct the searches:

((open-source wn ALL) OR (opensource wn ALL) OR (libre wn ALL) OR (OSS wn ALL) OR (FLOSS wn ALL))
AND
((safety wn ALL))
AND
((empirical wn ALL) OR (experiment* wn ALL) OR (case?study wn ALL) OR (survey wn ALL))*

The search string contains three main parts, separated by AND-clauses. The first part states that the article should include the term "open source" or some other synonym term that is often used, such as "OSS". The second part states that the article should include terms about safety. The third part makes sure that the article is empirical, by looking for terms like "empirical" and "experiment".

The '$\star$'-character is a wildcard representing any string of characters, which allows different grammatical numbers of the term to be identified, e.g. both "experimental" and "experimentation". The '?'-character is a wildcard representing one character, included because we want to identify both '-' and ' '. For example, both articles with the term "case study" and the term "case-study" are found. The term "wn" means that the phrase left of it should be found in the entity to the right of it, in this case ALL, which means all fields of database entries, such as title, abstract and key words. Text within {}-braces are searched as phrases and a search is not case sensitive.

In addition to the search performed on INSPEC and COMPENDEX databases using the described search string , all articles from the OSS-conferences (International Conference on Open Source Systems) from 2005 to 2013 [2] were inspected.

[1] http://jabref.sourceforge.net

[2] http://www.ifipwg213.org/

C. Exclusion criteria

After obtaining the initial result set of the articles by performing the search on the INSPEC and COMPENDEX databases, duplicate articles were removed. Subsequently, non-relevant articles were removed in two selection steps; first based on the title, keywords and abstracts, and then based on the full text. The exclusion criteria were defined during the design of the review protocol. We carried out the manual selection of articles based on the following criteria:

- Articles not about the usage of OSS, complete or partial, in the development of safety-critical systems were excluded from the selection.
- Articles about the usage of OSS for the security of critical systems were excluded from the selection, because the focus of our study is only on the safety aspects in safety-critical systems development.
- Articles about the usage of OSS for verification and validation of safety-critical systems were excluded. The focus in this study is on the development of safety-critical systems using OSS. Systems developed using closed source software but verified using OSS tools are considered out of scope.
- Articles about the usage of OSS for simulations of safety-critical systems were excluded from the selection. Simulation of systems is used for verification and validation activities. Again, it was not considered in scope of this study.
- Articles about the usage of OSS in the context of compilers and programming languages (e.g., research on type safety) were excluded from the selection. As described above the focus of the study is not to investigate the usage of open source tools for development of safety critical software. However, this could be seen as an interesting topic for further studies.

D. Selection of relevant articles

The aforementioned search query carried out on November 7, 2013., retrieved a set of 204 articles (see step 1 of Figure 1) which served as an input into the second step of this mapping study. Hence, upon downloading the title, keywords, abstract and author names of the articles in the retrieved set, it was identified that the set contained 63 duplicates out of which 35 were removed by the *JabRef* reference manager and 28 by manual search (step 2 of Figure 1). In the third step, shown in Figure 1, the remaining set of 141 articles was manually inspected by the first author resulting in identification of 43 relevant articles. The third author validated the selected articles, which resulted in 32 relevant articles. In the step four of the analysis, the first and the third author conducted full text analysis of the manually selected relevant articles, and identified 18 and 21 relevant articles, respectively (see step 4 of Figure 1).

In the selection steps three and four (first based on the title, keywords and abstracts, then on the full text) the articles were grouped as following.

- **Relevant:** Articles that clearly fulfilled the inclusion criteria.
- **Not relevant:** Articles that are out of the scope of this study per the exclusion criteria.
- **Possibly relevant:** Articles that required further discussions among the authors to establish whether they are relevant for this study or not.

In the step 5, shown in Figure 1, the two sets of the articles identified as relevant in the fourth step were compared, and it was determined that there was a disagreement in 5 articles. A total of 17 articles were included in the both authors' sets. The 5 articles in question were rechecked and consensus between the authors was made to exclude 1 article [13], as it did not clearly deal with OSS, and to include the other 4 articles ((5), (8), (13), (22)). Consequently, the set of included articles now contained 21 articles.

After the five step selection process carried out on the articles in INSPEC and COMPENDEX databases shown in Figure 1, in the next, sixth step, the OSS conference proceedings were manually scanned for articles that match the inclusion and exclusion criteria. This resulted in one additional article (12), brining the total number of the identified relevant articles in this mapping study to 22.

Finally, as a part of the final, seventh step shown in Figure 1, the reference lists of the selected articles were inspected for further relevant articles, but this did not result in any additional articles.

E. Data extraction and synthesis

Three authors were involved in the data extraction. First, the first author extracted basic information about the primary studies, i.e., *publication forum*, *publishing year*, and the *domain*. Then, the third author validated the work of the first author, and continued by extracting the *type of study* (categorized as descriptive, exploratory, evaluative, or experience report), and the *OSS used* in the studies. Finally, the sixth author validated the work of the third author. There was little disagreement, and the outcome of the validation steps was resolved by a refinement of the categorizations.

We used the open source network Ohloh[3] to analyze the identified OSS projects. We report the following characteristics from the Ohloh analysis: size of codebase in Lines of Code (LoC), number of contributors, number of commits, and main programming languages. Furthermore, we report a description of the codebase in terms of *maturity* (commit history of <1 year, 1-3 years, 3-5 years, or >5 years) and *activity* (inactive, very low, low, moderate, high, very high). Ohloh calculates activity relative to all OSS projects in the network, based on both contributor count and commit count. An OSS project is categorized as inactive if there has been no activity in the past two years. We extracted analysis results from Ohloh on December 9, 2013. For projects we could not analyze using Ohloh (e.g., no public URL to code repository available), we

[3] http://www.ohloh.net/

sent mails to the corresponding developers. Still, in some cases we could not extract the same information.

III. Validity assessment

During each step of this study (see Figure 1) special measures were taken to increase the validity of the research. In order to reduce the risk of incorrect removal of an article during the selection process, the first and third authors performed the selection independently that was further validated by the co-authors. The third step, shown in Figure 1, for the selection of relevant articles based on titles, keywords, and abstracts was validated by the third author by independently applying established inclusion and exclusion criteria. Similarly, the fourth step shown in Figure 1 for the selection of relevant articles based on full texts was carried out by the first and third author independently following the established criteria. The disagreement in the final selection sets was resolved by the co-authors by rechecking disagreed articles. After this, all the proceedings of the OSS conferences were manually checked, which resulted in one additional article. Then, the reference lists of all the selected articles were examined for more relevant articles. The data extraction, performed by the first and third authors, was also validated by the sixth author to improve the validity of the research carried out in this study.

By taking the above mentioned measures to enhance validity of this study we can be more confident that the majority of the relevant articles has been identified and included in the final articles' set.

IV. Results

We identified 22 primary studies that clearly indicate increasing research interest in OSS in safety-critical software systems; as 50% of the identified papers were published from 2011 and onwards. Further, most of the identified publications were published in conferences or workshops (18 papers, 82%), while only 4 articles (18%) were published in scientific journals. No primary publishing venue could be identified among the publications, however, a clear majority of the primary studies originate from domain specific conferences or journals, e.g., addressing topics such as robotics, fusion engineering, and intelligent transportation systems.

Table I shows an overview of the primary studies, which are also listed in Appendix. A majority of the studies are descriptive in nature, and present either OSS or systems integrating OSS. Three studies (5, 8, 9) report experiences from working with OSS in a safety context and one exploratory study discusses the potential of OSS in air traffic management based on a focus group meeting (11). Another exploratory survey discusses verification practices in biomedical OSS communities (14).

A. RQ1: Domains

As presented in Table I, the domains that are most represented among the primary studies are the automotive domain (5 papers), medical systems (5 papers), the nuclear domain (4 papers), and aerospace (3 papers). Previous surveys on compliance with safety standards [11] and evidence in safety certification [10] have also mentioned these four domains among the reported evidence. Interestingly, the aerospace domain represented in our study in only 3 papers was the top domain in both related surveys [10], [11]. At the same time, the most represented domain (automotive) in our study was not the most represented in the other two surveys (ranked the 4th and the 3rd) [10], [11].

Less explored, however still represented by primary studies, is work on OSS for safety-critical automation systems and maritime systems. Process industries and rail industry (mentioned in the top five domains in the evidence provided in both [10] and [11]) are not represented among the primary studies[4]. Finally, oil and gas, off-highway equipment and mining industries represented in the previous survey about compliance with safety standards [11] are not represented among our primary studies. Hence, it could conceivably be hypothesized that these industries have yet not been intensified by software systems or explored by open source solutions.

B. RQ2: OSS functionality

Few primary studies present *complete* safety systems that are *released as OSS in full* (see Table II). Paper (7) presents a system for slippery road detection, and paper (2) that describes the building management system SensorAct. Both these projects are however small in terms of both the codebase and the number of involved contributors, see Table III. CANOPNR (a Linux device driver) seems to have no commit activity for the last 12 months. The SensorAct (a middleware architecture) system has significantly more activity (75 commits in the last 12 months). It is possible, therefore, that these rather small systems that represent drivers or middleware with limited activity and changes do not challenge the safety-certification process and allow for full reuse of the entire codebase. However, more research on this topic needs to be undertaken before the association between the above mentioned factors is more clearly understood.

A majority of the primary studies instead describe safety-critical systems in which *OSS is integrated*. Table III presents the OSS systems that have been used in the primary studies. Several publications describe systems using operating systems that have been released as OSS ((1), (3), (5), (12), (15)) and used in automotive and medical domains. The integrated components include: Linux device drivers (1), communication framework (3), parts of the real-time operating system (5), Linux Debian distribution (12) and a cross-platform GUI library (15). Trampoline OS (5), Xen (12) and MicroGear (15) operating systems are more than five years old. Hence, it could conceivably be hypothesized that several years of presence in the OSS scene and solid history facilitate OSS operating systems integration into safety-critical systems.

Another type of OSS that has been repeatedly used implements imaging functionality. OpenCV, OSS for computer

[4] The IEC 61508 standard about safety has special variants for Automotive, Rail, Process industries, nuclear power plants and machinery, see http://en.wikipedia.org/wiki/IEC_61508.

TABLE I
OVERVIEW OF PRIMARY STUDIES. NAMES IN BOLD INDICATE THAT THE ENTIRE SAFETY-CRITICAL SYSTEM HAS BEEN RELEASED AS OSS.

Ref.	Author(s)	Year	Study type	Domain	Desc. of safety system	OSS used
(1)	Agafonovs et al.	2012	Descriptive	Automotive	Intelligent transportation system	OpenWRT, backports
(2)	Arjunan et al.	2012	Descriptive	Automation	Building management system	**SensorAct**, MongoDB, Quartz
(3)	Carvajal et al.	2013	Descriptive	Misc.	Ethernet communication framework	Atacama, NetFPGA
(4)	Centioli et al.	2005	Evaluative	Nuclear	Control system	RTAI
(5)	Choi	2011	Experience report	Automotive	Operating system	Trampoline OS
(6)	Dixit et al.	2011	Descriptive	Automotive	Drowsiness detection system	OpenCV
(7)	Enriquez et al.	2012	Descriptive	Automotive	Slippery road detection	**CANOPNR**
(8)	Falessi et al.	2005	Experience report	Misc.	Operating system	PVM
(9)	Gary et al.	2011	Experience report	Medical	Surgical Toolkit	IGSTK
(10)	Hall et al.	2009	Descriptive	Aerospace	Autonomous aircraft	Arduino, Paparazzi, MNAV, MicroGear, World Wind, OpenCV
(11)	Hardy	2006	Exploratory	Aerospace	N/A	N/A
(12)	Hu et al.	2009	Descriptive	Misc.	Operating system	Debian, Xen
(13)	Jan	2006	Descriptive	Aerospace	Positioning system	SBAS
(14)	Koru et al.	2007	Exploratory	Medical	N/A	N/A
(15)	Kuo et al.	2011	Descriptive	Medical	Wheelchair control system	Linux kernel, MiniGUI
(16)	Lu et al.	2012	Descriptive	Medical	Surgery navigation system	IGSTK
(17)	Pastore et al.	2010	Descriptive	Maritime	Autonomous surface vessel	MOOS-IvP
(18)	Sichta et al.	2007	Descriptive	Nuclear	Control system	EPICS, MDSplus, ACE, TAO
(19)	Toukabri et al.	2011	Descriptive	Automotive	Routing protocol	CarGeo6
(20)	Vaccarella et al.	2012	Descriptive	Medical	Sensor management	OpenCV, IGSTK, ACE, TAO
(21)	Van der Linden et al.	2008	Descriptive	Nuclear	Control system	MDSplus, HDF5, PyTables, ICE
(22)	Wang et al.	2010	Descriptive	Nuclear	Control system	EPICS

TABLE II
OVERVIEW OF SAFETY-CRITICAL OSS. THE LAST THREE COLUMNS REPORT RESULTS FROM AN ANALYSIS OF THE CODEBASE USING OHLOH.COM (DECEMBER 9, 2013).

Name	Ref.	URL	Desc.	LoC Main lang.	#Cont.; #Com.	Ohloh codebase description
CANOPNR	(6)	github.com/bitlessbyte/ CANOPNR	Slippery road detection	808; C++, C	1; 1	< 1 year history, very low activity
SensorAct	(2)	github.com/iiitd-ucla-pc3/ SensorAct	Building management system	7,034; Java	6; 159	1-3 years history, low activity

vision, has been used in both the medical and the automotive domains, see papers (6) and (20). OpenCV has over 5 years of history with a continuous and significant number of contributions. In the medical domain IGSTK (Image-Guided Surgery Toolkit) has been used in three primary studies ((9), (16), (20)). On the contrary, IGSTK despite over-five years history, has very low activity, see Table III. From the safety-certification perspective low activity (assuming that the code has high quality) should not be an obstacle but rather a facilitator since a limited number of changes do not trigger recertification.

In experimental nuclear reactors, three primary studies proposed using OSS. Two papers (18) (22) discussed using the control system EPICS (Experimental Physics and Industrial Control System), and one paper (21) used MDSplus (Model Data System). Both EPICS and MDSplus have more than 5 years of history and moderate or high activity, see Table III.

Communication and data management are two functionalities developed as OSS that were integrated in the medical (20) and nuclear domains (21). TAO OSS (20) has over five years of history and moderate activity while ICE (21) was not analyzed in this paper.

OSS for communication and data management are examples of two fundamental types of functionality that have been repeatedly integrated. Primary studies from both the medical (20) and the nuclear domain (21) have used available OSS for communication. Additionally, data management OSS solutions were integrated in the automation (2) and nuclear domains (18) (21). HDF5 data management OSS (21), MDSPlus (18) and MongoDB (2) have all over 5 years history and high activity.

To summarize, the review results indicate that required OSS functionality is more frequently integrated in the safety critical systems rather than the entire OSS solution is used. The complete OSS solutions found in the review are rather small and have low activity. On the contrary, OSS solutions *integrated* in safety critical systems have over five years of history and high or medium activity. Hence, we can conclude that long history of OSS projects facilitates their use in as integral parts of safety critical systems. However, the relationship between the activity of the OSS project and their adaption in safety-critical systems is less clear from the studied papers and therefore requires further investigation.

C. RQ3: OSS communities

The primary studies have integrated OSS of varying sizes (avg: 4,395,469 LoC, St.dev: 15,633,352 LoC). The largest OSS are by far Debian (76 million LoC, 6,248 contributors) used in various domains and the Linux kernel (16 million LoC, 11,714 contributors) used for the wheelchair control system in the medical domain, followed by OpenCV containing almost 3 million LoC (257 contributors) used in aerospace domain for the autonomous aircraft project and in automotive domain for sensor management.

Three other OSS projects contain more than 1 million LoC: OpenWRT, Paparazzi, and World Wind. OpenWRT was used in the automotive domain as a part of the intelligent transportation system. OpenWRT is the project with the most commits (81,638). World Wind, developed by NASA and used in the aerospace domain as a part of the autonomous aircraft system, stands out by having only 7 contributors. As indicated in Section IV-B small OSS projects are reused incompletely in safety-critical system while larger OSS projects are *integrated*.

Apart from Debian and the Linux kernel, the OSS project with the most contributors is Xen (458) used as an operating system. The number of contributions for the OSS projects integrated in safety-critical systems is much higher than OSS projects completely reused. This spin-off effect could be expected as a result of companies integrating OSS in their safety critical systems and after that sharing their adaptations, change and bug fixed in order to avoid double-maintenance costs.

Three OSS projects, apart from World Wind, have fewer than 10 contributors: CarGeo6, MiniGUI, and RTAI. The observed significantly lower number of contributions may be explained by low popularity of these OSS projects.

Figure 2 displays the current codebase activity and the commit history for the analyzed projects in Table III. Most OSS integrated in the primary studies have mature codebases, i.e., a commit history of more than five years. An exception is the small project CarGeo6, that has a short commit history of less than a year and no recent commits. Two other OSS projects, MITK and NetFPGA, are also inactive. Most OSS projects used in the primary studies are still active however, and often contain codebases with moderate to very high activity. In general, therefore, it seems that the relationship between the adaptation of OSS in safety-critical and their long history is rather clear. At the same time, further research should be done to investigate the relationship between the adaptation of OSS in safety-critical and OSS project activity. Moreover, it would be interesting to investigate the number of downloads of this project or its adoption within industry.

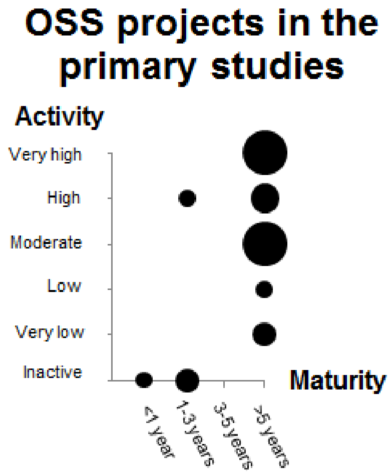

Fig. 2. Overview of codebase activity versus commit history (maturity) of the OSS projects in the primary studies. The circle areas represent the number of OSS projects. Note that the figure only contains the projects analyzed by Ohloh.com.

V. Conclusions

Overall it can be concluded that the number of publications in the area of open source and safety critical systems is increasing over time. This can be seen as an indication that the area is becoming more important and receives an increased amount of research.

Concerning the domains (RQ1) where research is conducted it was found that there are a number of expected domains where safety is important and where there also was found a number of papers. Examples of areas include medical devices, the nuclear industry, and the aerospace industry. However it was also found that there were some areas where safety is important but where no published research was found, such as in the areas of rail traffic, and the oil and gas industry.

Concerning the functionality (RQ2) provided by open source software it was found that more research report on systems that are subsystems of larger systems than complete systems. That is, only rather few papers were found that reported on open source software making up a complete system. Instead most published research concerns open source components that can be included in larger systems.

Concerning the open source communities (RQ3) that are investigated in research it was found that most were active and mature (> 5 years). However, there were examples of reported from communities that are no longer active.

As a final conclusion it can be seen that open source software today is used as part of safety critical systems. This is reported in both research that investigate complete systems and open source software components. It is reported from a number of different domains, and communities are active and mature.

References

[1] J. Feller and B. Fitzgerald, *Understanding Open Source Software Development*. Addison Wesley, 2002.

TABLE III

OVERVIEW OF OSS USED IN SAFETY-CRITICAL SYSTEMS. THE LAST THREE COLUMNS REPORT RESULTS FROM AN ANALYSIS OF THE CODEBASE USING OHLOH.COM (DECEMBER 9, 2013).

Name	**Ref.**	**URL**	**Description**	**LoC; Main lang.**	**#Cont.; #Com.**	**Ohloh codebase description**
ACE	(20)	www.cs.wustl.edu/ ∼schmidt/ACE-overview.html	Communication framework	497,733; C++	179; 72,366	>5 years history, moderate activity
Arduino	(10)	www.arduino.cc	Electronics prototyping platform	398,531; HTML, C, Java	59; 2,188	>5 years history, moderate activity
Atacama	(3)	No public URL	Communication framework	≈4,000; Verilog	5; ≈10	Not analyzed
backports	(1)	backports.wiki.kernel.org	Linux device drivers	- C	-	Not analyzed
CANOPNR	(1)	backports.wiki.kernel.org	Linux device drivers	- C	-	Not analyzed
CarGeo6	(19)	www.cargeo6.org	Network protocol stack	85,921; Autoconf, C	2; 11	>1 year history, inactive
Debian	(12)	www.debian.org	Linux distribution	76,183,250; C, C++	6,248; 575,249	>5 years history, very high activity
EPICS	(18), (22)	www.aps.anl.gov/ epics	Experimental physics and industrial control system	158,071; C, C++	47; 13,200	>5 years history, moderate activity
HDF5	(21)	www.hdfgroup.org/HDF5/	Scientific data management	644,319; C	53; 13,590	>5 years history, high activity
ICE	(21)	zeroc.com/ice.html	Communication framework	-	-	Not analyzed
IGSTK	(9), (16), (20)	www.igstk.org	Framework for image-guided surgery applications	99,755; C++, C	24; 1,756	>5 years history, very low activity
Linux kernel	(15)	www.kernel.org	Operating system kernel	16,281,512; C	11,714; 464,629	>5 years history, very high activity
MDSPlus	(18), (21)	mdsplus.sourceforge.net	Scientific data management	586,394; C, Java	12; 8,334	>5 years history, high activity
MicroGear	(10)	sourceforge.net/projects/ microgear/	Flight computer	- C++	-	Not analyzed
MiniGUI	(15)	sourceforge.net/projects hmgs-minigui	Cross-platform GUI library	101,069; C, C++, HTML	2; 396	>5 years history, low activity
MITK	(16)	mitk.org	Medical imaging interaction toolkit	567,021; C++, C,	139; 23,524	>5 years history, inactive
MongoDB	(2)	www.mongodb.org	NoSQL document database	575,424; C++, C	195; 23,944	>5 years history, very high activity
MOOS-IvP	(17)	oceanai.mit.edu/ moos-ivp	Library for autonomous vehicles	-	-	Not analyzed
MNAV	(10)	sourceforge.net/projects/ micronav/	Autopilot framework	- C, C++	-	Not analyzed
NetFPGA	(3)	github.com/NetFPGA	Hardware and software platform for research	203,459; coq, Perl, C	13; 421	1-3 years history, inactive
OpenCV	(6), (10), (20)	opencv.org	Computer vision and machine learning library	2,788,767; C++, HTML, C	257; 20,349	>5 years history, very high activity
OpenWRT	(1)	openwrt.org	Linux distribution for embedded devices	1,276,632; C, make, shell script	80; 81,638	>5 years history, very high activity

Paparazzi	(10)	paparazzi.enac.fr	Autopilot system for aircraft	1,269,686;
				XML, C
					80;
					19,748;
						>5 years history, Very high activity

PVM	(8)	Never released as OSS	Parallel virtual machine	-	-	-

PyTables	(21)	www.pytables.org	Scientific data management	201,336;
				Python
					31;
					7,193;
						>5 years history, moderate activity

Quartz	(2)	www.quartz-scheduler.org	Job scheduler for aircraft	52,151;
				Java
					37;
					1,626;
						>5 years history, Moderate activity

RTAI	(3), (4)	www.rtai.org	Real-time API for Linux	222,266; C	8; 1,752	>5 years history, moderate activity
SBAS	(13)	Not available	Internet server package	C	-	Not analyzed
TAO	(20)	www.theaceorb.com	Communication framework	966,093; C++	158; 42,688	>5 years history, moderate activity
Trampoline OS	(5)	trampoline.rts-software.org	Real-time operating system	650,593; C, XML, C++	18; 1,012	>5 years history, very low activity
World Wind	(10)	goworldwind.org	API for virtual globe (3D Earth)	1,141,236; C++, Java, C	7; 1,648	1-3 years history, high activity
Xen	(12)	www.xenproject.org	Virtual machine monitor	513,524; C, Python	458; 28,112	>5 years history, very high activity

[2] M. Höst and A. Orucevic-Alagic, “A systematic review of research on open source software in commercial software product development,” *Information & Software Technology*, vol. 53, no. 6, pp. 616–624, 2011.

[3] A. Bonaccorsi, D. Lorenzi, M. Merito, and C. Rossi, “Business firms’ engagement in community projects, empirical evidence and further developments of the research,” in *First International Workshop on Emerging Trends in FLOSS Research and Development, FLOSS’07*, 2007.

[4] J. West, “How open is open enough? melding proprietary and open source platform strategies,” *Research Policy*, vol. 32, no. 7, pp. 1259–1285, 2003.

[5] C. Ayala, D. S. Cruzes, A. D. Nguyen, R. Conradi, X. Franch, M. Höst, and M. A. Babar, “OSS integration issues and community support: An integrator perspective,” in *proceedings of the 8:th International Conference on Open Source Systems (OSS), Hammamet, Tunisia*, 2012.

[6] I. Sommerville, *Software Engineering, 7:th ed.* Addison Wesley, 2004.

[7] C. Lindholm, J. Pedersen Notander, and M. Höst, “A case study on software risk analysis and planning in medical device development,” *Software Quality Journal (electronic first)*, 2013. [Online]. Available: http://dx.doi.org/10.1007/s11219?013?9222?2

[8] B. Fitzgerald, K.-J. Stol, R. O’Sullivan, and D. O’Brien, “Scaling agile methods to regulated environments: An industry case study,” in *Software Engineering (ICSE), 2013 35th International Conference on*, 2013, pp. 863–872.

[9] N. G. Leveson, *Engineering a Safer World.* The MIT Press, 2011.

[10] S. Nair, J. L. de la Vara, M. Sabetzadeh, and L. Briand, “An extended systematic literature review on provision of evidence for safety certification,” Simula Research Laboratory, Tech. Rep., 2013.

[11] S. Nair, J. L. de la Vara, M. Sabetzadeh, and D. Falessi, “Management of evidence for compliance with safety standards: A survey on the state of practice,” Simula Research Laboratory, Tech. Rep., 2013.

[12] B. Kitchenham and S. Charters, “Guidelines for performing systematic literature reviews in software engineering,” *Technical Report Keele University and University of Durham*, vol. 2.3, 2007.

[13] A. Gribovskiy, J. Halloy, J.-L. Deneubourg, and F. Mondada, “Building a safe robot for behavioral biology experiments,” in *2012 IEEE International Conference on Robotics and Biomimetics, ROBIO 2012 - Conference Digest*, Guangzhou, China, 2012, pp. 582 – 587. [Online]. Available: http://dx.doi.org/10.1109/ROBIO.2012.6491029

APPENDIX: PRIMARY STUDIES

(P1) Agafonovs, N.; Strazdins, G.; Greitans, M., Accessible, customizable, high-performance IEEE 802.11p vehicular communication solution, In Proceedings of the International Conference on Nuclear Engineering, 2010, 6, 495 - 499.

(P2) Arjunan, P.; Batra, N.; Choi, H.; Singh, A.; Singh, P.; Srivastava, M. B., SensorAct: A privacy and security aware federated middleware for building management, In Proceedings of the 4th ACM Workshop on Embedded Systems for Energy Efficiency in Buildings, 2012, 80 - 87.

(P3) Carvajal, G.; Figueroa, M.; Trausmuth, R.; Fischmeister, S., Atacama: An open FPGA-based platform for mixed-criticality communication in multi-segmented ethernet networks, Proceedings - 21st Annual International IEEE Symposium on Field-Programmable Custom Computing Machines, FCCM 2013, 2013, 121 - 128.

(P4) Centioli, C.; Iannone, F.; Mazza, G.; Panella, M.; Pangione, L.; Podda, S.; Tuccillo, A.; Vitale, V.; Zaccarian, L., Optimization of RF power absorption by optimization techniques using the lower hybrid current drive of FTU, Fusion Engineering and Design, 2005, 74, 543 - 8.

(P5) Choi, Y., Safety analysis of Trampoline OS using model checking: An experience report, In Proceedings of International Symposium on Software Reliability Engineering, ISSRE, 2011, 200 - 209.

(P6) Dixit, V. V.; Deshpande, A.V.; Ganage, D., Face detection for drivers’ drowsiness using computer vision, IFMBE Proceedings, 2011, 35 IFMBE, 308 - 311.

(P7) Enriquez, D.J..; Bautista, A.; Field, P.; Kim, S.-i.; Jensen, S.; Ali, M.; Miller, J., CANOPNR: CAN-OBD programmable-expandable network-enabled reader for real-time tracking of slippery road conditions using vehicular parameters, 15th International IEEE Conference on Intelligent Transportation Systems, 2012, 260 - 264.

(P8) Falessi, D.; Pennella, G.; Cantone, G., Experiences, strategies and challenges in adapting PVM to VxWorks hard real-time operating system, for safety-critical software, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2005, 3666 LNCS, 209 - 216.

(P9) Gary, K.; Enquobahrie, A.; Ibanez, L.; Cheng, P.; Yaniv, Z.; Cleary, K.; Kokoori, S.; Muffih, B.; Heidenreich, J., Agile methods for open source safety-critical software, Software - Practice and Experience, 2011, 41, 945 - 962.

(P10) Hall, C. J.; Morgan, D.; Jensen, A.; Chao, H.; Coopmans, C.; Humpherys, M.; Chen, Y. Q., Team OSAM-UAV’S design for the 2008 AUVSI Student UAS Competition, Proceedings of the ASME Design Engineering Technical Conference, 2009, 3, 575 - 584.

(P11) Hardy, J.-L.; Bourgois, M., Exploring the potential of OSS in Air Traffic Management, Open Source Systems, Springer US, 2006, 203, 173-179.

(P12) Hu, Y.; Long, X.; Wen, C., Virtual Machine Based Hot-Spare Fault-Tolerant System, 1st Int. Conference on Information Science and Engineering, 2009, 429 - 432.

(P13) Jan, S.-S., Providing corrections from space based augmentation system to non-aviation users via internet, In Proceedings of the 19th Int. Technical Meeting of the Satellite Division, 2006, 2925 - 2929.

(P14) Koru, G.; El Emam, K.; Neisa, A.; Umarji, M., A survey of quality assurance practices in biomedical open source software projects, Journal of Medical Internet Research, 2007, 9, e8 (40 pp.).

(P15) Kuo, C.-H.; Syu, Y.-S.; Tsai, T.-C.; Chen, T.-S., An embedded robotic wheelchair control architecture with reactive navigations, 2011 IEEE International Conference on Automation Science and Engineering, 2011, 810 - 15.

(P16) Lu, T.; Liang, P.; Wu, W.-B.; Xue, J.; Lei, C.-L.; Li, Y.-Y.; Sun, Y.-N.; Liu, F.-Y., Integration of the Image-Guided Surgery Toolkit (IGSTK) into the Medical Imaging Interaction Toolkit (MITK), Journal of Digital Imaging, 2012, 25, 729 - 737.

(P17) Pastore, T.J..; Patrikalakis, A.N.., Laser scanners for autonomous surface vessels in harbor protection: Analysis and experimental results, 2010 International Waterside Security Conference (WSS), 2010, 6 pp.

(P18) Sichta, P.; Lawson, J.; Mastrovito, D.; Roney, P.; Tindall, K., Preliminary design of NCSX central computing and control, In Proceedings of the Symposium on Fusion Engineering, 2007.

(P19) Toukabri, T.; Tsukada, M.; Ernst, T.; Bettaieb, L., Experimental evaluation of an open source implementation of IPv6 GeoNetworking in VANETs, 11th Int. Conference on ITS Telecommunications, 2011, 237 - 245.

(P20) Vaccarella, A.; Enquobahrie, A.; Ferrigno, G.; De Momi, E., Modular multiple sensors information management for computer-integrated surgery, Int. J. of Medical Robotics and Computer Assisted Surgery, 2012, 8, 253 - 60.

(P21) Van der Linden, G.W..; Wijnoltz, F.; Scholten, J.; Busch, P.J..; Poelman, A.J..; Smeets, P.H..; de Groot, B.; Koppers, W.R.., Design of the Magnum-PSI safety, control and data acquisition system, Fusion Engineering and Design, 2008, 83, 273 - 275.

(P22) Wang, H.; Li, G.; Cao, L., Concept design of computer monitoring system for ITER transfer cask system, In Proceedings of the International Conference on Nuclear Engineering, 2010, 6, 495 - 499.