A Theoretical and Practical Assessment of Open Source Development and Dissemination of Construction IT Solutions

T. Hartmann

Assistant Professor, University of Twente – P.O. Box 217 7500 AE Enschede, The Netherlands – t.hartmann@ctw.utwente.nl

N. Hartmann

US Patent Agent, DeVries & Metman - World Trade Center Arnhem, Nieuwe Stationsstraat 10, 6811 KS Arnhem, The Netherlands – n.hartmann@dvme.nl

ABSTRACT:

Many Construction IT systems do not adequately support the complex and frequently changing business processes on architectural, engineering, and construction (AEC) projects. To improve this situation there is the immanent need for AEC researchers to explore IT development and dissemination strategies that are based on technology pull theories that allow AEC practitioners to influence the development of the IT solution before, during, and after its dissemination. One such technology pull development and dissemination strategy is Open Source. The paper shows how the Open Source model is theoretically in the position to overcome some of the inherent problems of the AEC industry with adopting and using new Construction IT solutions. The paper further analyzes a number of existing Open Source project management software projects to assess the potential to support AEC practitioners. The paper concludes with a strategy to establish an Open Source AEC community.

1 INTRODUCTION

In the next decades the Architectural, Engineering, and Construction (AEC) industry needs to provide environmentally and socially sound infrastructure for the growing world population. To cope with this rising demand AEC engineers require advanced IT systems that are able to flexible support their complex and frequently changing work processes. Unfortunately, today's general purpose IT systems do not support the business processes of the AEC industry well. This is mainly because IT development and dissemination management models are based on technology push theories that assume that inventors need to first develop IT systems before innovators can diffuse them strategically through a population of possible adopters. Unfortunately, however, such IT solutions do not allow users to easily and quickly adjust the solution to changing work processes as it is required by AEC practitioners. Thus, there is the immanent need for researchers to explore IT development and dissemination theories that are based on technology pull theories that assume that potential users can define and influence the development of the IT solution before, during, and after its dissemination.

Open Source software development and management models are an example of models that are based on such technology pull theories. Theoretically, users can choose among a large number of possible applications, and quickly adjust applications to changing project environments. In other words, the Open Source model allows users to pull IT solutions according to specific requirements.

This paper intends to give a first theoretical overview about the possibilities of the AEC industry to leverage on Open Source software. To do so, the paper first drafts a theoretical argument of why the Open Source model is poised to overcome the problems of the AEC industry to adopt new information technologies. The paper then reports the results of an analysis the potential of a number of existing Open Source projects to already support AEC practitioners. Finally, the paper concludes with a tentative strategy of how the AEC industry can establish a strong Open Source community by considering a number of managerial, financial, and legal issues.

2 PROBLEMS WITH CURRENT SOFTWARE DEVELOPMENT AND DISSEMINATION

Most of today's information technology (IT) management models focus on the development of large standardized IT solutions and how to disseminate these solutions through as large adopter networks as possible. These management models average out the variability of different possible user scenarios (Carroll 2000). The general management philosophy is that users should adjust to IT systems and not IT systems to users. IT system users have little influence on the development of the systems and little possibilities to adjust systems to specific and changing work processes. It is not surprising, that technology push IT developments and disseminations have been largely successful in developing and disseminating IT solutions that support common and mainstream work processes of large industry spanning user groups, such as Office Applications. However, technology push IT development and dissemination models have not been very successful to support the work processes on complex projects with many unique and changing requirements, such as new product development projects, construction projects, or even many academic research projects.

One industry that mainly operates on such complex projects is the AEC industry. The industry is characterized by a lack of IT systems to specifically support the complex work processes on projects. So far, the development and dissemination of IT solutions with mainstream technology push models has failed.

We attribute this failure to two problems of the current software development and dissemination process. First, from the perspective of AEC companies, it is often not feasible to adopt general purpose IT systems that are developed under technology push models because

- Users cannot adjust these IT systems flexibly to changing requirements of their work processes.
- The development and dissemination of IT systems requires large upfront investments in the training of users and in changing work processes to match the IT system. Companies that frequently need to adjust IT systems to changing work processes, such as project based AEC organizations, cannot justify such investments.

Second, from the perspective of IT providers, it is not feasible to develop IT systems using technology push models because of the temporary and drifting character of construction projects that makes it impossible for most IT development firms

- To research all information upfront that is needed to implement new general purpose IT systems to support a large number of users across different types of AEC projects and AEC companies,
- To push general purpose IT solutions through a large group of potential users that would allow the generation of sufficient revenue to justify the necessary upfront investments, and
- To offer services which allow the needed timely and frequent adjustment of the IT system's functionality to the varying AEC work processes.

To overcome these problems, researchers need to find new ways to develop and disseminate IT solutions that are based on technology pull models. Such methods are able to provide state of the art IT systems that users can easily adjust to local and varying project environments and that do not require large amounts of upfront investments before practical implementations. One such technology pull IT development and dissemination model is Open Source software management.

IT developed with the Open Source software management method has already been very successful in developing and disseminating easily adjustable IT infrastructure systems, such as web servers (Apache), operating systems (Linux), or database servers (MySQL). Theories of how to establish the needed thriving Open Source communities exist and have worked in practice. The next section briefly describes these extant theories about Open Source projects.

3 OPEN SOURCE – A TECHNOLOGY PULL IT SYSTEM DISSEMINATION AND DEVELOPMENT APPROACH

Open Source software has its origins in the early days of computing when communities around hobby programmers evolved who freely exchanged code with others. Soon some of these hobby programmers realized that some of their code has value for others and funded firms to sell executable versions of their programs. These firms considered their source code as their major intellectual property and started to protect the code physically and legally against others who intended to copy it. The proprietary software model was born that dominated the computer industry for the years to come. However, the Open Source software model never completely deceased and developed many powerful software solutions, such as the Linux operating system, the Apache web server, or the MySQL database application. What distinguishes Open Source software from proprietary is not the fact that the software is freely available, many proprietary companies offer their software as so called freeware, but that the underlying programming code of the software is openly available (Watson et al. 2008). This enables anybody to download the source and change it to their liking if they wish to do so.

The open availability of the source code further allows a potential large number of volunteers and non-volunteers to contribute to the development of specific software (Mockus et al. 2002). Open Source software projects usually do not work with an explicit assignment of programmers to tasks, but programmers choose the tasks they would like to work on. Further, usually there exists no explicit system or detail level design, and no project plans, schedules, or predefined deliverables to organize the programming activities of community members. Despite these missing explicit project management tools, Open Source communities were able to develop sophisticated software tools. Furthermore, these successful communities are characterized by a relatively stable group of programmers, for example, the half life of contributors for the Debian operating system project is around 6.5 – 7.5 years (Michlmayr & Robles 2008) and these programmers on average increase their involvement over time. Such a constant and voluntary involvement seems to enable Open Source developers to work with a higher productivity than proprietary programmers. Mockus et al. (2002) showed, for example, that the productivity of the programmers of the Apache project was 1.5 times higher compared with other proprietary projects. Further, Mockus et al. (2002) claim that Apache programmers write code with more care and creativity.

The long commitment of Open Source programmers to their projects combined with their large number allows Open Source projects to rapidly respond to problems and to flexibly adjust the solutions to varying and constantly changing work processes. Further, it allows Open Source projects to find defects in their software more quickly. This, in turn, allows for a significant minimization of the downtime of the software in case users detect deficiencies while using the software. Further, the overall defects within the software seem to be lower as well, if compared with proprietary software. Mockus et al. (2002), for example, found that the defect density of the Apache software was far less than the one of comparable proprietary products.

In summary, the Open Source software development and dissemination method has been reportedly able to maintain software projects with a quality of code that was at least comparable with the quality of proprietary projects, while at the same time Open Source projects were able to leverage on the benefits of mass innovation (Watson et al. 2008). Additionally, Open Source communities were able to mitigate two strategic risks better than proprietary companies: they do not have the risk involved with a wrong assessment of the demand and the related need to price the software adequately, and they were able to mitigate the innovation risk by drawing on an immense pool of talented developers with in depth knowledge about the projects' source code.

4 THE POTENTIAL OF AN OPEN SOURCE PLATFORM FOR THE AEC INDUSTRY

It is not surprising, that in recent years, products published under Open Source software licenses have become major forces on the software market in the areas of Operating systems, and web and database servers. Next to these applications to support the IT industry with their tasks Open Source projects theoretically have the potential to support the work tasks of the AEC industry. This is mainly because Open Source software development projects have, at least theoretically, several advantages during the dissemination over traditional proprietary software development. First, the Open Source business model does not operate with software licensing costs (Watson et al. 2008). Therefore, Open Source software can support AEC companies with their traditionally limited IT budgets. No large upfront investments are necessary for AEC companies to initiate the first use of the software. In this way, Open Source software applications can overcome the initial barriers that many AEC companies face when they are deciding upon whether they should improve their internal work processes around a specific IT system or not.

Once an AEC company made the decision to use an Open Source IT system, Open Source software further enables professionals to easily and dynamically adjust the IT system to the unique project contexts the AEC industry works in. Well-managed Open Source software suites are comprised of several small "modules" that together form a package. Each of the modules performs a particular task and can be dynamically added or removed from implemented IT solution. Software developers can easily develop new packages and integrate them in already implemented IT systems. Further, this modular structure allows it that programmers quickly and easily can change modules without jeopardizing the

Table 1: The study's Open Source projects and their key statistics and overall all-time downloads

Name	Type	Members	Downloads	Active since
BRL_CAD	CAD	43	693'780	23.03.2004
FreeCAD	CAD	9	293'782	17.03.2002
Archimedes CAD	CAD	9	95'337	12.07.2005
Avocado CAD	CAD	5	37'583	19.01.2007
Gantt Project	PM	6	3'014'861	29.01.2003
Open Proj	PM	15	1'270'423	23.06.2007
Gantt chart based project management	PM	1	38'901	1.12.2002
Plandora	PM	4	11'967	13.07.2004
Notepad ++	Control	5	15'515'433	24.11.2003
AsphaltOpen	Control	1	32	31.08.2008

functionality of other modules. In this way, Open Source software theoretical supports the ongoing adaptation to shifting user requirements that is so necessarily needed to support complex AEC projects.

5 AN ASSESSMENT OF THE CURRENT STATE OF OPEN SOURCE AEC SOFTWARE

5.1 *Research Method*

To access the current state of Open Source software to support AEC practitioners we used quantitative data from the Open Source repository Sourceforge.net (http://www.sourceforge.net). Sourceforge.net is a web based repository that serves as a centralized location for software developers to control and manage Open Source projects. According to Wikipedia (http://www.wikipedia.org), Sourceforge.net hosts more than 180,000 Open Source projects and has more than 1.9 Million registered users. Official web statistics show that the sourceforge.net domain attracts more than 28 million visitors annually.

We started our data collection effort by identifying adequate Open Source projects with relevance for the AEC industry. Hereby, we focused on two different overall application types: two dimensional (2D) and three dimensional (3D) Open Source Computer Assisted Design (CAD) applications and project management scheduling applications. We decided to focus on these two applications types as they represent two main software applications that are used by engineering designers and construction project managers, two of the main groups of AEC practitioners.

To identify suitable CAD applications we searched the Open Source projects on Sourceforge.net that have been indexed by the project owners with the keyword "CAD". This query resulted in 213 projects overall. From these 213 projects we chose the 4 projects that Sourceforge users had downloaded most often and that, according to their project description, represented general-purpose CAD projects that were in an active development status. We for example, sorted out the "QCad" project, because it is no longer developed, or the "Linux Drawing Viewer project" and the "Sailcut CAD" project as they do not represent general-purpose CAD applications.

Similarly, we identified the most downloaded Gantt chart project management applications by searching all Sourceforge.net projects with the keyword Gantt. This search identified 61 projects, from which we again chose the 4 most downloaded projects that represented general purpose Gantt chart scheduling applications.

Finally, we also identified two non AEC related software applications, one ranked among the top 20 all time downloaded applications, and one project that was just recently started. We used these two

Table 2: Key Developer and User Statistics of the Projects from May 2008 to April 2009

Name	Source Read	Source Write	Help Forum	Open Forum	Download
BRL_CAD	181'395	4107	262	51	341'980
FreeCAD	12'131	4812	465	328	118'822
Archimedes CAD	N/A	N/A	2	10	46'371
Avocado CAD	454	24	N/A	48	19'136
Gantt Project	604	305	367	157	867'600
Open Proj	8'132	592	343	113	834'675
Gantt chart based project management	0	0	0	0	0
Plandora	4'128	439	9	2	3'589
Notepad ++	8'270	422	2799	3963	6'859'911
AsphaltOpen	8	45	0	0	31

projects during data analysis as a benchmark with which to compare the analysis results of the AEC projects.

For each of these projects, we collected data that is freely available on the Sourceforge.net website that allowed me to analyze the programming, but also the user activity of each of the projects. In particular, we collected data about the number of developers of each of the projects. We collected the amount of total downloads since the start of the project, and the amount of downloads from May 2008 to April 2009. We collected how often the project's software developers changed the project's underlying source code between May 2008 and April 2009. Finally, we also collected the number of posts to the general help user forum of each of the projects between May 2008 and April 2009.

5.2 *Results*

We conducted an initial analysis of these data to establish the current state of Open Source projects with the potential to support the AEC industry. The number of downloads (Table 1) indicates that there are some promising projects already in wide use. The most promising packages are the CAD applications "BRL_CAD" and "FreeCAD ", and the scheduling applications "GanttProject" and "OpenProj".

Further, the data showing how often developers read the packages' source codes and wrote back changes to the source code between May 2008 and April 2009 shows that developers worked actively on these four projects in the last year. Further, there seemed to be a vibrant user community as indicated by the traffic in the Help and Open forums during this time period.

We also conducted time series analyses of the download, developer, and forum activities to look at possible trends within these. A graphical analysis of the graphs in Xxx reveals no significant trends of any of the Open Source packages that we studied. Only the "OpenProj" project shows some increase in the number of downloads between August 2008 to October 2008. Further, xxx shows an increase in the help forum activity of the "FreeCAD" project between September 2008 and December 2008. Overall, however, all projects showed relatively stable user and development activities throughout the time period under study.

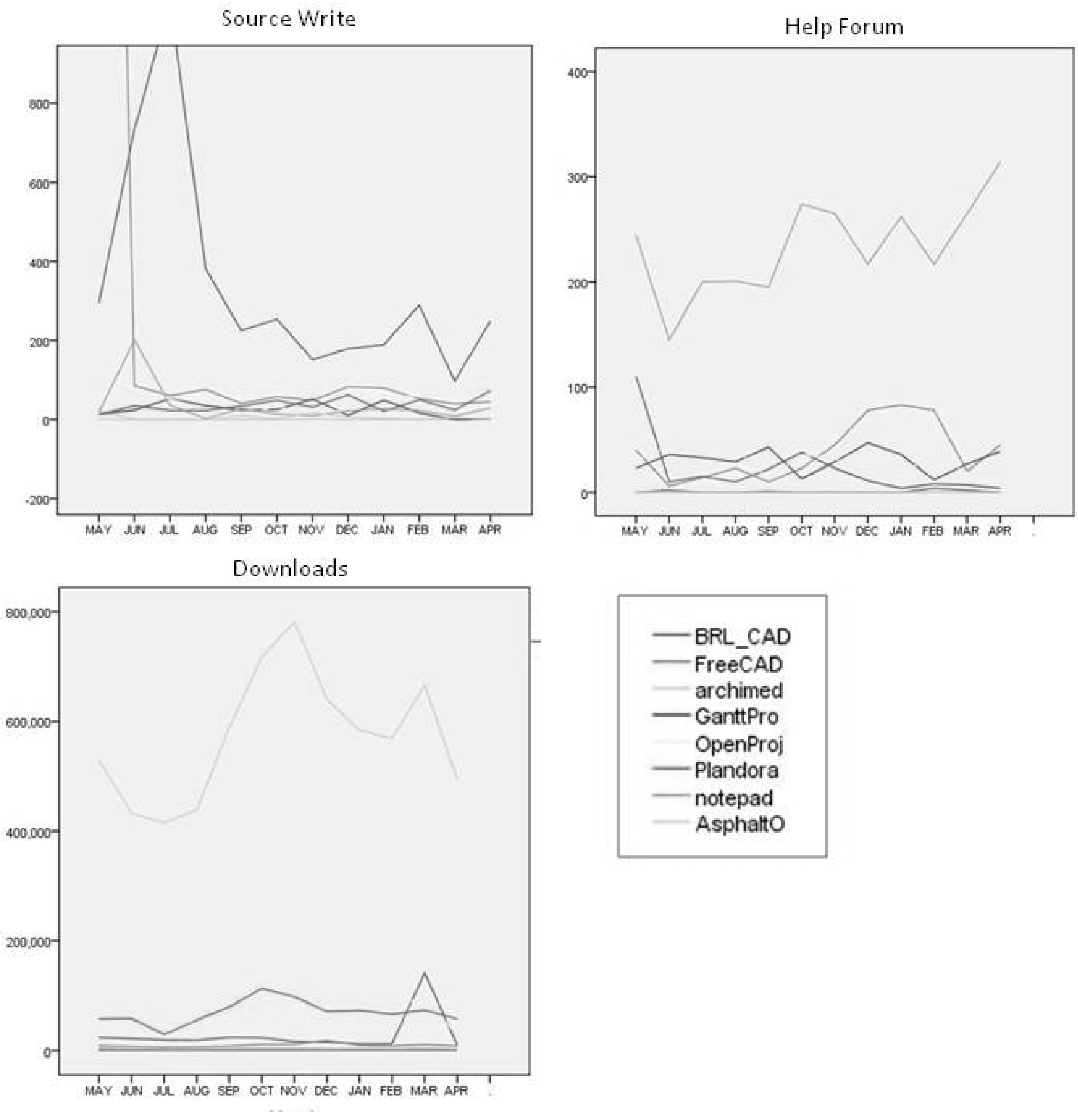

Figure 1: Time series of selected key project data from May 2008 to April 2009

5.3 *Discussion*

In summary, the data we collected on Sourcefore.net shows four relatively promising Open Source packages with the potential to support the AEC industry. These packages are the two CAD packages BRL CAD and FreeCAD , and the two project management scheduling packages GanttProject and OpenProj.

According to the overall number of downloads and, in particular, according to the number of downloads in the last year, all these four packages seem to be in frequent use. Additionally, the development and user activity around these four projects as indicated by the trend graphs of downloads, source code write activities, and help forum entries, was relatively stable. Therefore, one can generally conclude that the packages work well for users and that there seems to be sufficient support through updating the software frequently and by discussing the software in the respective forums.

The question remains how much of the projects' activity is, however, driven by members of the AEC community. It is most likely that, for example, software managers, who use the software to schedule their software development efforts, drive most of the activities around the two scheduling packages. Similar, it is likely that the two CAD packages are used mainly by non-AEC engineers. The question remains, therefore, how the AEC industry can leverage the potentials of these specific packages and of Open Source software overall. In the next section, we outline a first strategy of moving forward in this direction, by drawing on previous innovation management research that observed how successful Open Source communities have been established and managed in the past.

6 PROPOSED FUTURE STRATEGY TO ESTABLISH A STRONG AEC OPEN SOURCE COMMUNITY

In this section, we propose a strategy of how we believe the AEC industry can move forward to leverage on the theoretical potentials of Open Source software. In the first part of this section, we will describe the strategy of how to establish a thriving AEC Open Source community, in the second and third part of this section, we will briefly discuss some implications with respect to financial considerations, and patent and legal property laws.

6.1 *Establish an AEC Open Source Platform*

We propose that developing an Open Source AEC community should start with establishing a core management group to drive the following developments. This core group should at first focus on establishing a common platform that provides links, documentations, and forums that are targeted specifically to AEC practitioners and how AEC practitioners can use the existing Open Source projects. At the same time the core management group should work on establishing software programmers with an AEC background in the different core developer groups of the existing Open Source software projects. These AEC developers can then start driving the further development of the packages to support specific functions that the users of the packages with an AEC background request.

Once the core management group has established an initial momentum of the AEC Open Source community, it is important to establish sub-groups with different responsibilities. As mentioned the asynchronous communication of AEC community members requires relatively small working groups with less than 15 members (Au, 2009). Once the work load for the (voluntary) members of the core group gets too high, it is, therefore, important to establish a number of sub-groups responsible to support parts of AEC industries software needs. These new groups could be, for example, a CAD group, a scheduling group, and a cost group. Later on, the community can establish further satellite groups such as a 4D group that works on providing software at the interface between CAD and schedule, a quantity take off group that works on providing software at the interface between the CAD and the cost group, or a cash flow and resource allocation group that works on providing Open Source software at the interface between cost and schedule.

After establishing the satellite groups the main responsibility of the core group should switch to ensuring the interoperability between the different software packages that the sub-groups support. Hereby, the core group can establish best standard programming guides, define interfaces and work on integrating these interfaces into the different software packages. Further, all groups of the AEC Open Source software community should motivate cross group memberships of their members. Such cross-group membership will ensure a specific focus on the interoperability between the different groups' packages and cross-group members will be an important source of knowledge to develop and implement exchange standards. Finally, research on other Open Source projects has shown that cross-group membership increased the development efficiency of cross-group developers (Au, 2009).

In the long, the AEC Open Source community should focus on establishing a complete AEC Open Source eco system around its Open Source project. Such an eco system should be comprised of independent consultants, authors, publishers, and, last but not least, educators.

6.2 *Financial Considerations*

One general misconception of Open Source is the belief that Open Source projects solely rely on volunteers to conduct the necessary development and management work. Many companies, such as RedHat, MySQL, or JBoss developed business strategies around Open Source projects that enabled these companies to hire full time staff to work on and support their Open Source projects (Watson et al. 2007).

In general, there are three possible sources of funding to support the AEC Open Source community. The first source of funding comes from AEC companies that have an interest in using Open Source software to support their employee's work processes. These companies could fund the AEC community in general, but also specific software development efforts to particularly ensure that the Open Source applications provide the companies' employees with specific required functions. Next to financial support, AEC companies can also dedicate some employees to contribute actively to the Open Source community during their work time.

The Open Source community can secure a second source of funding by providing consulting and educational services to interested parties. Such services could include the development of company-wide implementation strategies, the support of specific projects, the organization of training courses, and the publication of books and other training material.

The third possibility to secure funding that is also the most promising possibility to support the start of the AEC Open Source community beyond the work of volunteers, is the possibility to apply for publicly or privately available research grants. The work on an AEC Open Source community offers tremendous research possibilities in the areas of organizational science, engineering informatics, and computer science that justify the funding of efforts around the AEC Open Source community through research grants.

Researchers can, for example, make contributions to the area of organizational science by exploring the feasibility of Open Source software to support the non-technology-savvy user groups or the AEC industry, how social networks of non-technology-savvy users can assemble around Open Source communities, or how Open Source can improve the management of IT implementations in complex industries. Researchers in the area of engineering science can explore how Open Source systems can support complex engineering work processes in the construction industry and other industries. Further, this area of research also offers tremendous possibilities to help researchers get started with efforts to develop prototypes because the Open Source AEC community will offer a code infrastructure that allows researchers to easily integrate their new ideas into the existing code base. Through such integration, these research prototypes become available to a large audience and, in this way, there are tremendous possibilities to increase the impact of the worldwide research around developing new technologies for the AEC industry. Finally, researchers can contribute to existing computer science theory, by exploring ways of how software engineers can develop more flexible and adaptable IT solutions to support technology "pulls" of non-technology savvy users.

6.3 *Legal Considerations*[1]

There is a hot debate about the legal position of software projects with respect to patentability, copyright, and enforceability. Both US Patent Law and European Patent Convention state that computer programs "as such" are not patentable. However, computer programs may be the subject matter of a patent when a court or a patent examiner does not regard the computer program being "as such." Thus, computer programs are not excluded from patentability under all circumstances. Various case law decisions have already identified under which circumstances computer programs may be considered patentable (see, for example, T208/84, T26/86, T209/91, T6/83 from the Boards of Appeal of the European Patent Office). As a result, by now quite a substantial amount of software patents exist and there is a real risk that particular software infringes one or more of these existing patents.

This risk is present not only for organizations and individual programmers, but also for users and distributors of software. However, the argument that Open Source software is more vulnerable than proprietary software because it allows anybody to directly view the source code, thereby making it easier to determine what is and what is not included in the code for the purpose of establishing potential patent infringement is no longer valid. Current technologies allow determining whether or not a binary program of a proprietary software package includes particular source code. Thus, the risk of infringing a patent is no less for proprietary software than it is for open source software.

Even further, there are arguments that Open Source communities face fewer legal risks than proprietary software companies. First, patent holders are not likely to sue individual programmers or small organizations associated with software projects. The reason for this is that individual programmers and small organizations do not hold substantial assets or receive income from which the patent holder can obtain significant royalties (Richard Fontana et. al., 2008). Commercial distributors of the software are the more likely targets of the infringement suits. Second, and even more importantly, Open Source code is legally considered as part of open domain knowledge. Once published Open Source code may no longer be a subject matter of a patent filed at a later date. Therefore, the Open Source projects do not face the risk that another party obtains a patent covering the already written code and subsequently demands royalties.

To mitigate the legal risks, the AEC Open Source community should maintain a detailed documentation of the published source code and have an infrastructure in place that can track and confirm the release dates of the source code into the public domain. Doing so may help prove that the source code of the AEC Open Source community formed a part of published prior art before the earliest filing date of the patent, thus possibly invalidating the patent on which this code could otherwise infringe. Documenting and tracking code in this way is anyway necessary to allow for the distributed team work that lies at the heart of Open Source code development. Therefore, the introduction of code documentation and tracking practices does not pose a big obstacle for an Open Source community. In addition, the AEC Open Source community may officially establish a corporate umbrella organization

[1] Next to the cited references, this section is based on the practical experiences of the second author working as a certified patent attorney.

to protect its members from personal liability for the code they developed.

According to Fontana et al. (2008), trying to find and evaluate all of the existing patents that may cover the subject matter of the community's source code upfront is not advisable. First, there are simply too much previous patents to search for. Second, searching patents requires special searching techniques. Non skilled searchers, such as software programmers, usually assume that patents found are broader in scope than these patents actually are which causes them to become overly or unnecessarily worried. Third, properly deciphering the legal patent language to determine whether or not particular source code may infringe the claims of the patent requires the participation of a skilled patent attorney. Finally, if a party is aware of the existence of a particular patent, the party may be a subject of increased damages if they continue to infringe a patent despite their knowledge of it.

Down the road, when the AEC Open Source community becomes well established with a stable source of financial income, the community should consider hiring certified attorneys to properly address legal issues on copyright, patenting and trademark on a project by project basis.

7 CONCLUSION AND OUTLOOK

This paper theoretically argues for the potential of Open Source software to support the dynamic work processes of the AEC industry. We theoretically argue how Open Source software through its modular character, but also through the availability of programming resources can overcome much of the problems of the AEC industry to cope with the customization problems the industry has currently with existing proprietary software. Additionally, we theoretically argue that Open Source software due to its minimal overall cost of ownership has the potential to overcome the problems that the many small AEC companies of the fragmented AEC industry have with the implementation of software tools to support their work processes.

Building on these theoretical arguments the paper offers a first assessment of existing Open Source projects with the potential to support AEC work processes. The paper closed with describing a strategy of how the authors think that the AEC industry could establish a vibrant Open Source community, considering managerial, financial, and legal aspects.

Using this strategy we expect that the AEC industry will be able to benefit from the theoretical possibilities that an Open Source software development and dissemination promises. In this way AEC companies can improve their productivity. This improved productivity, in the long run, will be a big stepping stone in helping the AEC industry to leverage the potential IT applications offer to improve the AECs work processes and productivity. This will help the industry to cope with the challenges it will face in the next decade to satisfy the worldwide needs to provide affordable infrastructure and housing to a growing global population. Furthermore, the potential of IT systems to support AEC products and processes will improve the quality of the built environment. An improved quality of our built environment, in turn, will have significant impacts of the environment and on the way the societies of this world can live together.

8 REFERENCES

Au, Y.A., Carpenter, D., Chen, X., & Clark, J.G. 2009. Virtual organizational learning in open source software development projects. Information & Management, 46(1), 9-15.

Carroll, J.M. 2000. Making Use: Scenario-Based Design of Human-Computer Interactions. MIT Press Cambridge, MA, USA.

Fontana R., Kuhn B., Moglen E., Norwood M., Ravicher D., Sandler K., Vasile J., Williamson A. 2008. A Legal Issues Primer for Open Source and Free Software Projects. Version 1.5.1. Software Freedom Law Center. New York.

Michlmayr & Robles 2008. Volunteers in Large Libre Software Projects: A Quantitative Analysis Over Time. In: Sowe, S.K.S.I.G., & Samoladas, I.M. (eds.) 2008. Emerging free and open source software practices. Hershey, PA, USA: IGI Publishing.

Mockus, A., Fielding, R.T., & Herbsleb, J.D. 2002. Two Case Studies of Open Source Software Development: Apache and Mozilla. ACM Transactions on Software Engineering and Methodology, 11(3), 309-346.

Sowe, S.K.S.I.G., & Samoladas, I.M. (eds.) 2008. Emerging free and open source software practices. Hershey, PA, USA: IGI Publishing.

Watson, R.T., Boudreau, M.-C., York, P.T., Greiner, M.E., & Wynn Jr, D. 2008. The Business of OPEN SOURCE. Communications of the ACM, 51(4), 41.