

Tool Usage in Students' Software Projects

Pekka Mäkiäho and Timo Poranen

School of Information Sciences, University of Tampere
33014, University of Tampere, Finland
{pekka.makiaho, timo.t.poranen}@uta.fi

Abstract

Software tool selection and usage play an important role in software development. Selecting, and then using, right tools makes development process faster and thus the product quality will get better. In this paper we evaluate tool selection process and tool usage in 14 students' software development projects. Data was collected from final reports and from three questionnaires filled by team members during the projects. On average, the teams used fourteen different tools during their projects. Main reason to select a tool was previous experience of one or more team members. Most common reason to change a tool to another was that not all team members were committed to use the first selected tool.

1.0 Introduction

Software tools are always needed in software development to implement new software. There are hundreds of tools that can be used to develop software in different platforms and environments. Selecting the right tools for a software development project is often a hard task and requires a lot of attention from the project management and organization's technical support. Problems with the usage of development tools are also very common project risks [1,2].

Software project courses are usually taught in all universities teaching computer sciences [3]. In these courses a group of students design, implement and test a software product. Depending on the course organization, students may quite freely select what tools they use in their project.

In this paper we investigate what kinds of tools are used in students' software development projects. We also analyze what the selection and change processes are and what kind of difficulties students encounter with the tools.

In Section 2 we give an overview on tool usage and tool classification in software development projects. In Section 3 we describe how data was collected. In Section 4 we analyze tool selection and change processes, and what kinds of tool related risks the teams encountered. Finally we discuss future research directions.

2.0 Tools in Software Development Projects

A common categorisation for project risks is to divide them into six groups: technology risks, people risks, organisational risks, tool risks, requirements risks and estimation risks [1]. Tool risks are closely connected to developers' skills to use the selected tools.

Tools vary between different platforms, programming languages and programming environments. New tools are published regularly, and existing tools are given updates, and their versions and features change rapidly. Therefore it is understandable that software developers encounter tool related problems in their work.

Tools that are used in software development and other software development project activities are often called CASE (Computer Aided Software Engineering) tools. Tools can be classified based on their function. Some possible tool categories are planning tools, editing tools, change management tools, configuration management tools, prototyping tools, method-support tools, language-processing tools, program analysis tools, testing tools, debugging tools, documentation tools and re-engineering tools [4].

Portillo-Rodríguez et al. [5] listed over 130 tools that are used in global software development. The authors classified the tools into 13 different categories, (virtual meeting tools, quality tools and socio-cultural tools among others).

Fuggetta [6] proposed a classification of CASE tools into three categories based on how they support software development processes. The first category is tools that support individual process tasks (like code compilation), the second is workbenches that support process phases (like design), and the third is environments that support all, or at least a substantial part, of the development process.

3.0 Data gathering

During the academic year 2011-2012, 30 MSc. students and 67 BSc. students were divided into 14 software development projects at the University of Tampere, School of Information Sciences [7]. Project teams were formed based on students' earlier studies and wishes. MSc. students acted as project managers and BSc. students participated as developers in the projects.

Initially, all projects had 2-3 project managers and 5-6 developers. All project topics were different, and their target platforms were either www-applications (10 projects), mobile phone applications (3 projects) or desktop applications (1 project). Only one of the applications used network to communicate with another application. The average project size was 1331 hours [7]. All projects applied agile or iterative development models.

The students were free to select what tools they use in their projects. The only tools provided by the university were Subversion for version control and Redmine for project management; however, it was not compulsory to use these tools. In some projects the client's organisation also provided development tools.

During the course all students were asked to answer questions in the Moodle forum of the course on their personal tool usage. The first questions were given two months after the start of the project. The students were asked to list all the tools they had used and report what the purposes of these tools were.

In the middle of the projects, the students were asked to report any changes in tool usage (new tools, ceasing to use a tool, or changes in usage frequency). Questions on tool usage changes were also asked at the end of the projects.

In Table 1 the project type, the used project management tools and the used development tools are listed. The number in the parenthesis indicates how many of the team members used that tool.

Table 1: Project types and usage of project management and development tools.

	Project type	Management tools	Development tools
1	Mobile	Redmine (6), Kanbanery (5)	Eclipse (4), Android SDK (1), RapidSVN (1)
2	WWW	Redmine (4)	Eclipse (3) ApacheSVN (2), Jenkins (1), Microsoft Visio (1)
3	WWW	Redmine (7)	NetBeans (7), Joomla (5), Putty (1), Filezilla(2), Python (1), ApacheSVN (1), Dreamweaver (1)
4	WWW	Jira (4), DokuWiki (4)	NetBeans (4), TortoiseSVN (3), XAMPP (2), Putty (1), PHPMyAdmin (1)
5	WWW	Redmine (8)	Eclipse (6), PHPMyAdmin (2), Joomla (2), RapidSVN (1), ApacheSVN (3), Zend Server (2), XAMPP (1), Subclipse (1)
6	Mobile	Redmine (5)	Eclipse (7), Android Development Tools (5), Adobe Device Central (1)
7	WWW	Jira (6)	Putty (4), Navicat (2), PHPMyAdmin (2) NetBeans (7), TortoiseSVN (3), MySQL (2), Firebug (2), Xampp (2)
8	WWW	Redmine (6)	GitHub (6), NetBeans (5), Vim (1), SSH (1), XAMPP(1), MySQL (1)
9	WWW	Redmine (6)	Eclipse (6), MySQL (4), Subversion (6)
10	WWW	Redmine (1), GoogleWiki (7)	MySQL (1), PHP Symphony (2), Visual Studio (1)
11	Mobile	Redmine (6)	Tortoise SVN (3), Visual Studio (4), Windows Phone SDK (1)
12	Desktop	Redmine (4)	ApacheSVN (3), Struts (1), Editplus (1), Tomcat (1), Netbeans (5), MySQL (2)
13	WWW	Redmine (5)	Opera Dragonfly (1), NetBeans (3), Eclipse (4), ApacheSVN (2) MongoDB (1), Morphia (1), Tomcat (1)
14	WWW	Mediawiki (4)	Abobe flash (3), TortoiseSVN (3)

In Table 2 the communication tools, document sharing tools and the other tools used in the projects are presented.

Table 2: Communication, document sharing and other used tools.

	Communication	Sharing and Documentation	Other tools
1	Gmail (5), IRC (5), Skype (1), SMS (1), Phone calls (1)	Googledocs (4)	Sttackoverflow (1), PhotoShop (2)
2	Skype (4), Gmail (2)		ConTEXT (1), Vim (1)
3	Facebook (5), Gmail (2)	Open Office (1), GoogleDocs (7)	Joomla's support-forum (1), Notepad++ (2), SciTE (1), Microsoft Office Picture Manager (1) Photoshop (1)
4	Flowdock (4), DokuWiki (4), GMail (1)	Mediawiki (1), Googledocs (3), MS Office (1), Libre Office Writer (1)	Poedit (1), Notepad++ (1), Pencil (1)
5	Facebook (8), IRC (1), Tomboy notes (1), GMail (1), Doodle (1),	Google Docs (6), LucidChart.com (2), MS Visio (1), Mediawiki (2), MS Word (1), OpenOffice Writer (1)	Evernote (1), gedit (1)
6	Flowdock (6), Gmail (3), Google Calendar (1)	Google Docs (4)	
7	Gmail (1), IRC (3)	Mediawiki (6)	CakePhp cookbook (1), Dia (1), ConTEXT (1)
8	IRC (3), ThunderBird (1)	Google Docs (4)	
9	Flowdock (6), Skype (1), GMail (1)	Horde (1), LibreOffice (2), OpenOffice.org Writer (2)	Emacs (1), Notepad++ (1), Photoshop (1)
10	Gmail (3)	Google Docs (1)	Google Git (2), Qt Creator and Designer (6), Photoshop (1), Blender (1), Gimp (1), Silo 3D (1), 3DCoat (1), Notepad++ (1)
11	Thunderbird (1), GMail (1), Skype (6), IRC (1), TeamViewer (1)	Microsoft Word (1)	Notepad (1), Photoshop (1), Resharper (1)
12	Doodle (1), Skype (3), GMail (3)	Mediawiki (4), MS Office (1), GoogleDocs (3)	Axure RP (2), UltraEdit (1), Notepad (1), Balsamiq (1), Photoshop (1)
13	IRC (1)	LucidCharts (2), Google Draw (3) GoogleDocs (2), Dropbox (1)	Balsamiq (3)
14	GMail (4), Messenger (3)	Google Docs (4)	Paint (1), Photoshop (2), Adobe Illustrator (1)

Since the projects were relatively small, we decided to classify used tools into previously mentioned five categories: project management, development, communication, sharing and documentation, and other tools.

Table 3: Tool selection and tool change process.

	Selection process	Change reason	Other findings
1	Own evaluation, team member's recom., client's recom., course recom.	Not all used the tool, a more suitable tool was found.	In the end of the project SMSs and phone calls were also used. IRC was replaced with phpBB forum.
2	Course recom., earlier experience	No formal decisions	
3	Course recom., manager's recom., earlier experience, team member's recom.	A tool did not work as expected, new needs came.	SVN did not work as expected and thus was the process was changed and FTP was taken into use.
4	Manager's recom., earlier experience	Not suitable technology to use a tool, a more suitable tool was found.	Managers selected the tools they could provide help and also tools of which knowing could benefit in working life.
5	Earlier experience, free to use, to try a new tool, manager's recom.	Not all used the tool, a more suitable tool was found.	IRC was replaced by Facebook – the decision was made together.
6	Own evaluation, manager's recom., earlier experience, course recom., closely related to project topic	No formal decisions	Using Android tools was natural due to the platform
7	Manager's recom., earlier experience, closely related to project topic	No formal decisions	
8	Client's recom., own evaluation, earlier experience, to try a new tool	No formal decisions	Moodle was given as a framework by the client. The group used lot of time on choosing the right tools.
9	Manager's recom., team member's recom.	No formal decisions	
10	Client's recom., manager's recom., team member's recom.	No formal decisions	
11	No process	Not all used the tool, a more suitable tool was found.	IRC was replaced by Skype as it was easier to use.
12	Team members recom., manager's recom.	No formal decisions	The tools were chosen by majority and it seems that all were not happy on that.
13	Course recom., earlier experience, closely related to project topic	No formal decisions	
14	Earlier experience, client's recom.	No formal decisions	

In addition to the three tool usage questionnaires, the project managers were asked questions on the tool selection and change processes. At the same time with the first questionnaire, the managers explained their tool selection process while the change processes were reported simultaneously with the second questionnaire.

The purpose of these questions was to find out how the teams selected their tools and how and why they made significant changes in their tool usage.

Tool selection processes and reasons to change tools, with possible comments and explanations, are shown in Table 3. From the data it was easy to detect different reasons behind the tool selection. The reasons were categorised as follows: a specific tool was recommended by a team member, manager, client or the course; someone in the team had previous experience on using a certain tool; the team wanted to try and learn a new tool; a certain tool was closely related to the used platform (like Android development tools in an Android project), or the team evaluated several tools to find the most suitable for the project. One team also reported that they selected a particular tool because it was free to use.

After the projects were finished, the teams wrote their final reports. In one section of this document the encountered risks were discussed. Table 4 lists both the foreseen and unforeseen tool risks that were realized.

Table 4: Realized foreseen and unforeseen tools risks.

	Foreseen tool risks	Unforeseen tool risks
1	Emulator performance issues	Android not being as open as we thought it is, Trying Kanban without physical board, one computer broke.
2	To learn Joomla took time	Technical problems with Joomla.
3	Not reported	Not reported
4	Not reported	Not reported
5	Not reported	Not reported
6	Not reported	Not reported
7	Access control on CakePHP was challenging.	Web-socket technique was unforeseen technology which the team wasn't able to adopt adequately enough.
8	Not reported	Project members did not use Redmine on reporting as often as managers wanted.
9	Not reported	Eclipse was not familiar tool for one member.
10	Not reported	Lack of development tools, as the development environment set relatively high requirements for the hardware.
11	Not reported	Not reported
12	All team members had not common platform experience	Not reported
13	Group members did not have previous experience on MongoDB nor REST	Not reported
14	Not reported	Selection of an unfamiliar programming language caused a slow start.

4.0 Analysis

Next we analyse more specifically the data given in Tables 1-4. We do not provide references for the tools because more information on them can be found easily using web search engines. On average, the teams used fourteen different tools during their project.

4.1 Usage of project management and development tools

The total number of different project management (PM) tools used in all projects was four if we consider the different wikis (GoogleWiki, DokuWiki and MediaWiki) as one tool. The most frequently used project management tool was Redmine. This is no surprise given that it was provided and recommended by the university. Redmine was used in 11 projects out of 14 by 53 students. In nine projects, it was the only project management tool, and in two projects it was used together with another PM tool.

Jira was used in 2 projects by 10 students. One project used Jira as the only PM tool and the other one together with DokuWiki.

A wiki was used in 3 projects by 15 students. In one project it was the only PM tool, in two projects a wiki was used with another PM tool (Redmine or Jira).

As regards the usage of PM tools, it seems that in some projects not all team members were committed to use the selected PM tool(s). In projects 2, 4, 12, and 14 only four team members reported that they used the team's PM tool. It is also possible that some team members simply failed to report on their tool usage correctly.

On average, the teams used 4.7 different development tools. Altogether the number of different development tools used by the projects was 32. The most frequently used tools were Eclipse with 29 users, NetBeans with 26 users and MySQL with 10 users. These were used in 5 different projects.

Although the teams did not explicitly report that they used version control, it can be seen from the list of development tools that almost all the teams used some version control system with a client application, or some integrated development environment with a direct connection to the version control.

Out of the 32 development tools, 21 were used in only one singular project. 14 of these were not used by more than one student.

It can be seen from Table 1 that there was more variation on the usage of development tools than on project management tools. The Redmine project management tool was used in 11 projects while the three most common development tools were used only in 5 projects. The main reason for this was that Redmine was recommended in the course.

4.2 Usage of Communication, document sharing and other tools

In addition to phone calls, short messages and emails, the students used 10 different software tools for communication. The most frequently used tools were IRC, Skype, Facebook and Flowdock. Altogether 57 students used these 4 tools. The other 6 tools had only 12 users.

As regards documentation and document sharing, the most common tool was GoogleDocs with 38 users in 10 different projects. The total number of users for all other documentation and sharing tools was only 34.

There were also 28 other tools that did not fall into any category. Typically these tools were used only by 1-2 students. An exception was Adobe Photoshop which was used in 6 projects – only one project member in each project, though.

The students were free to select the tools they used. Moreover, in most projects, the project managers did not create any policy on tools. So, it is natural that many different tools were used especially in the category “Other tools”. On the other hand, as regards communication and document sharing, it is important that inside the project, all members use the same set of tools. This is why the number of tools is smaller in those categories.

4.3 Tool selection and change process

As mentioned, the groups were free to use any tools they needed. The course recommended the students use Redmine. In some projects the client also gave recommendations, and in some cases there were not really so many alternatives available due to the nature of the project.

Personal experience had the most significant effect on tool selection: 9 groups out of 14 reported “earlier experience” as a reason for selecting a certain tool. The second most important reason (7 groups) was “manager’s recommendation”.

In some groups the managers dictated the tool set while in other groups the tools were selected together by discussing or brainstorming. Only one project reported that they did not have any selection process.

It is noteworthy that more than half of the projects (8/14) did not report any official changes in tool usage during the project. However, even in those projects, certain tools were dropped and other tools were taken in use, but no common decisions were made.

The most frequently reported reasons for tool changes were “not all used the tool” (3) and a “more suitable tool was found” (4). In all cases where the communication tool was dropped, the reason was that the tool was not used by all project members, and therefore it was replaced with a more suitable tool. For example, IRC was not familiar to everyone; Skype was not installed in all work stations etc.

4.4 Tool related risks

Only 12 tool related risks were reported in these 14 projects. Five of these were foreseen (mentioned in the project plan) and seven unforeseen. Out of the five foreseen risks, three were related to inexperience of project members and two were technology issues.

In the category of unforeseen risks, two risks realized because of wrong technology expectations and two because the users were not familiar enough with the tools selected.

5.0 Conclusions

In this paper we analysed the tool usage of 14 software development projects. Our focus was on getting an overview on what kinds of tools are used in projects. Findings of this paper can be used by the course teachers to emphasise importance of the tool selection process and to show sample tool sets used in earlier projects to the project teams. In future, the correlation of tool selection and tool usage with the process and product quality can be studied more precisely.

6.0 References

- 1 Hughes B and Cotterell M, Software Project Management, 5th edition, McGraw-Hill, 2009
- 2 Ahtee T and Poranen T, Risks in students' software projects, 2008, in Proceedings of CSEE&T 2009, 4 pages
- 3 Ahtee T and Poranen T, Teaching software projects in universities at Tampere, in Proceedings of INSPIRE XII: Improving Quality in Computer Education (editors: Berki, E., Nummenmaa, J., Sunley, I., Ross, M. and Staples, G.), pages 87-101, 2007
- 4 Sommerville, I., Software Engineering 8, Addison-Wesley, 2006
- 5 Portillo-Rodríguez J, Vizcaíno A, Piattini M, Beecham S, Tools used in Global Software Engineering: A systematic mapping review, Information and Software Technology, Volume 54, Issue 7, July 2012, Pages 663-685
- 6 Fuggetta A, A classification of CASE technology, Computer, Volume 26, Issue 12, pages 25-38, 1993
- 7 Mäkiäho P and Poranen T (editors), Software Projects 2011-2012, University of Tampere, School of Information Sciences, Report 2, http://www.uta.fi/sis/reports/index/R2_2011.pdf (visited November 2011)