Caring and sharing computer files with configuration management

Andrew P. Wandel, Michael D. Jokic, and Alexander A. Kist
School of Mechanical and Electrical Engineering, University of Southern Queensland
Corresponding Author Email: andrew.wandel@usq.edu.au

Structured abstract

BACKGROUND
Group-work projects are commonly included in engineering courses to prepare the students for the workplace where this is the standard situation. One of the difficulties students face is that they do not have access to a unified file system where they can create and edit documents related to the project. Such a system reduces the risk of members duplicating work because someone was working offline for a substantial period, makes the integration of individuals' contributions simpler since all the work is visible and keeps the latest version of a document visible to all members. While the internet cloud has recently offered some solutions (e.g. Dropbox), a genuine Configuration Management solution is superior because it allows such features as merging different versions of a file and ensuring that only one person is able to edit a file at any given time: cloud solutions may allow multiple instances of file editing and therefore only stores the last instance of saving—deleting the changes made by the first person. Furthermore, Configuration Management systems store copies of each revision made to the files, so that it is possible to revert an individual file to a previous instance if it is found that subsequent changes were erroneous, unnecessary or undesired.

PURPOSE
To investigate the impacts the implementation of a comprehensive Configuration Management solution has on student behaviour with regard to file-sharing practices and how that affects the overall student experience, thereby helping achieve the learning outcomes.

DESIGN/METHOD
A configuration management system was set up so that each team could share files with each other and the staff also had access. Usage of the system was made "compulsory" by allocating a few marks based on the quality of the system usage. Surveys were conducted following the 2011 and 2012 offerings of the course to assess the students' opinions on the operation of the system.

RESULTS
Most students saw that the benefits in easily managing their team's files outweighed the administrative cost of setting up and running the system. They also appreciated the teaching team having direct access to their files without having to actively share the files (which can be frustrating if the student shares a subset of files and omits a file which is critical).

CONCLUSIONS
Configuration Management systems are an industry standard that can improve student project outcomes by providing an environment where students are able to automatically manage their file sharing in a consistent and coherent fashion that protects against the errors that occur when manual administrative systems are implemented. The large majority of students appreciated the benefits of this system and were willing to devote the small amount of initial time required to setup the system because of the overall advantages, so this intervention resulted in positive student behaviour and outcomes. The staff also found it easier to manage the facilitation of the team project by having direct access to the students' files.

KEYWORDS
Configuration management, teamwork, file sharing

Introduction

Configuration management is a set of principles and practices that is used to track the changes made to project artefacts and products, with many artefacts being computer files. It is therefore a vital component in quality assurance processes in the delivery of products or designs so that they can be certified under ISO 9001 or CMMI (Capability Maturity Model Integration). Various standards have been created which specifically address the design of a configuration management system for best practice, including STANAG 4159 (NATO, 1991), MIL-STD-973 (USDoD, 1992), ISO10007:2003 (ISO, 2003) and IEEE Std. 828-2005 (IEEE, 2005). Configuration management is often supported by specific software tools.

The primary aspect of configuration management is "version control", whereby a file goes through a process of modification; each iteration ("revision") is readily identifiable as that file, but distinct from all other iterations. At a basic level, version control is commonly performed by renaming the file to indicate some information about who changed it, when it was changed or why it was changed. This is normally with the original filename preserved and appropriate codes or description appended, with the final result causing the visible file system (the "folder") to display all revisions thereby requiring searching to identify which is the active version of each file.

At the other end of the spectrum, a system is implemented whereby the filename is preserved, but multiple copies (for each revision) are stored in the background so that only the active ("live") copy is visible in the file system. To access a different revision to the live copy requires entering the system, but the file system remains uncluttered. However, it becomes necessary to "lock" the active copy (prevent the file from being edited by more than one person at a time) in order to prevent a saving conflict. This conflict is colloquially known as the "last person to save wins" scenario, where the first person to finish their editing produces a live copy that is overwritten by the last person to finish, deleting any changes that had been made by others.

Another aspect of configuration management is file distribution. The *ad hoc* method would have each user store a working copy on their local drive and whenever they make changes, they must distribute the changes to anyone who requires that updated version. This creates problems because it relies on someone who has made a new revision to identify everyone who potentially would need access to the file and for all recipients to store the new version properly so that when they need to access it (for information or editing purposes), they are accessing the correct version. One solution is to have the files stored on a central drive with users opening them remotely, i.e. the file is edited using software on the user's (local) machine, but no copy of the file is maintained on the local machine. A file-system based implementation of this would be to have a network drive with users having permission to read/edit various folders on that drive.

An example of a web-based implementation is Microsoft Sharepoint, where an internet browser is used to navigate through a file structure. Another common solution to this problem is to have the files stored on a local drive, but have any new revisions shared with a server that then broadcasts the new revision to all linked users (e.g. Dropbox, Google Docs or the "Cloud"). The primary advantage of the file-system solution is that it is relatively straightforward to lock a file because most programs will recognise when a file has been accessed for editing and therefore prevent someone else from making different changes at the same time. The primary advantage of the web-based solution is that it is relatively cheap and simple to implement.

More detail on the importance of configuration management and how it is implemented has been reported (Hinsen, Laufer, & Thiruvathukal, 2009; Spinellis, 2005). Most importantly for the undergraduate context, it has been recognised that because configuration management is commonly used in industry, it ought to be a tool that graduates have utilised in an educational context (Broman, Sandahl, & Baker, 2012). Reid and Wilson (2005) utilised the

tool *CVS* for assignment submission in computer science, with the conclusion that the earlier in the degree programme that a student is exposed to configuration management tools, the better they are able to utilise it. Broman et al. (2012) incorporated configuration management as one aspect of a capstone group project course, which was essential since each group contained approximately 30 members. In the current paper, the authors investigate the impact of introducing a configuration management system in a second-year group project course. "Distributed" teams, where the members live and work in different locations around the world, are becoming increasingly common and the effects this has in business contexts have been studied for some time (McDonough, Kahn, & Barczak, 2001; Pinto, Pinto, & Prescott, 1993). The ability to manage file distribution and versioning becomes more important in such situations and is pertinent to the current study because the cohort contains a large proportion (approximately two-thirds) of distance-education ("external") students.

Current work

The configuration management (CM) system was implemented in the course ENG3103 Engineering Problem Solving Computations as a pilot in 2010. After an analysis of available systems, a SubVersion server was used to host the repositories and TortoiseSVN was recommended as the client interface to the repository. This enabled many of the introductory problems to be resolved ready for full implementation in 2011, when it was made compulsory for all students to use. Previously, the main methods students had available to them for sharing files (with each other and the teaching staff) were: email, forum posts, USB drive (if their team could meet face-to-face) and a wiki (which was very unreliable) on the Learning Management System (LMS). The primary reasons for introducing the system were:

- Inefficiencies in sharing files. The file sharing process was not transparent: copies (different versions) of the same file could appear in multiple threads of the forum. Unless a student was diligent in saving the file each time it appeared in a post, they could easily be working on a different version to others. This problem is exacerbated when emails are used as a supplementary vehicle, because if the email was not broadcast (sent to all members of the team), then some members of the team only had access to a superseded version of the file. The ultimate example of how this was detrimental to the students was when one team submitted the wrong version of the final report. (Ironically, the same happened after introducing the CM system, so human error can still play a part! But the students were quickly able to find the version they should have submitted.)
- Inefficiencies in editing files. Students tend to do most of their work close to the deadline, so when there is a report where different members are responsible for producing the content for the various sections, it can be difficult to collate all the content. Someone might make changes that are detrimental to the overall product, someone might make changes that others do not know about and it is likely that at some point multiple people are working independently on the same document. Combining this input is a tedious task, particularly if they are unaware that another is editing that file.
- Inefficiencies in assisting the students. For the teaching staff, compounding the aforementioned problems were that students would often request that we "look at file X" without attaching the file to the email/forum post, which either entailed searching for said file (and hopefully finding the correct version) or replying with a request for that file, which delayed the delivery of assistance. These problems were exacerbated because of the nature of the course (learning and using computer programming), so that if the code failed to work correctly with the error occurring when calling a certain file, the students tended to send only that file. Unfortunately, many errors are caused by previous operations in other files that only become problems later on, so requests for (all) other files were common.

Configuration Management systems address these issues by:

- Having a file structure that is replicated across all members of the team (i.e. each member sees identical folders and files contained therein). When someone has added or made changes to a file, this information is sent to the server, which propagates the change to others when they contact the server. This guarantees that everyone has access to the latest version of the files. This operation is similar to the manner in which email systems work: there is a server somewhere which stores the master copy of the emails for an account. When a user accesses their account (whether by using a client such as Outlook or Thunderbird, or using a web interface such as logging in after navigating to a url in an internet browser), the server sends the updated information (new emails). While the user is connected, any changes made (deleting, sending emails) are sent to the server for processing.
- Allowing files to be locked, i.e. the person who was first to open the file can prevent others from editing the file (they could potentially open a read-only version). This guarantees a sequential process of editing so that every change is stored. This process avoids the "last person to save wins" effect that occurs with some solutions (e.g. Dropbox), where if two people have downloaded a copy of a file and make changes to it, the file that is kept is by the last person to connect to Dropbox and commit the changes. (Note that Dropbox resolves this problem by creating a new file with a message as part of the filename to preserve the file that was over-written.)
- Keeping track of which user made changes to the file. Essentially, the server stores every copy of a file that is "committed" (uploaded) to it. This means that users can know who has worked on the file when.
- Allowing old versions of the file to be retrieved. Because the server stores every committed copy, if it is found that a change has been detrimental, the file can be "reverted" to an old version to become the working copy.
- Allowing concurrent changes by different users to be merged into one file. For trivial changes this can often be done automatically.

The teaching staff were aware that by offering a solution to address the difficulties, other problems would be introduced, namely the learning curve to become familiar with the software and its use, which can often be seen as another burden imposed on the students. Overall, it was felt that the benefits to the long-term efficiency of the teams was worth the initial difficulties in learning the system. One of the purposes of this paper is to report whether the students felt similarly after using it.

Method

This paper reports on the outcomes of implementing the CM system in the 2011 and 2012 offerings of ENG3103. The students were surveyed anonymously at the end of semester, with the survey made available after submitting their final team report for assessment and some respondents completed the survey after receiving their mark for that assessment. Table 1 contains the number of responses for the survey; the rates are relatively high, which provides some confidence as to the validity of the results.

The first consideration is whether the students appreciated the system overall and Figure 1 shows that the overwhelming majority of students did so. Note that quotes are included verbatim, except where meaning is lost.

> Respondent 5 (External, Agree): *having all the files in one place, being able to lock files for editing to avoid doubling up*
>
> Respondent 8 (External, Strongly Agree): *at meetings we could access all data using one computer and work on things together. easily see progress on setions of assignments and know what step would be good to look at*

Table 1: Number of survey responses; response rate for each group in brackets.

Year	On-campus (Internal)	Distance-Education (External)
2011	19 (19.6%)	28 (23.7%)
2012	15 (21.4%)	18 (13.7%)

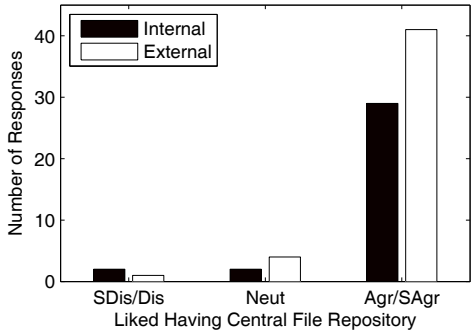

Figure 1: Student responses to "I liked having a central repository for all our files".

> Respondent 13 (External, Agree): *I liked that the files were safe from any computer errors. It was useful to be able to see where other team members were up to and to guage their contribution without needing to trawl through a backlog of emails. All participants had and oppertunity to edit the same document.*
>
> Respondent 23 (Internal, Strongly Agree): *The previous method of collaboration was to attach files to posts on study desk, but this ended up in a huge mess of heaps of different versions of different files. Configuration Management:*
> *1. Kept everything centrally accessible*
> *2. Could be instantly updated if changed*
> *3. Ensured that you weren't overwriting important old files*
>
> Respondent 24 (External, Strongly Agree): *Easy access to files made our project much easier than trying to upload them in somewhere like Wiki or Team Forums. Quick to update if you need another team member to review. Kept track of history and revisions so you could always revert back.*
>
> Respondent 49 (Internal, Agree): *Easy to revert changes. Good to see contributions of other team members as well as any changes made.*
>
> Respondent 51 (External, Strongly Agree): *It makes team work easy.*

The feeling was obviously not universal (three students responded "Disagree or Strongly Disagree"):

> Respondent 20 (External, Strongly Disagree): *nothing really came out as outstanding. it is a central repository and there are better real life systems to manage projects smartplant, documentum etc*
>
> Respondent 35 (External, Neutral): *Get rid of it all together. Students email and skype each other and therefore are able to transfer files easily. This system makes the coarse that more confusing and time consuming.*
>
> Respondent 78 (Internal, Disagree): *It was a waste of my time.*

To investigate the causes of this negativity, the correlation between the students' response shown in Figure 1 and two questions which assess the students' capabilities with software are shown in **Error! Reference source not found.**. From these results, it can be seen that all the students who did not like having the central repository find it difficult to learn new software and those who were ambivalent or actively negative towards the concept of having the central repository did not find it easy to use. It would appear that the major hurdle for positive engagement is making the setup as seamless as possible.

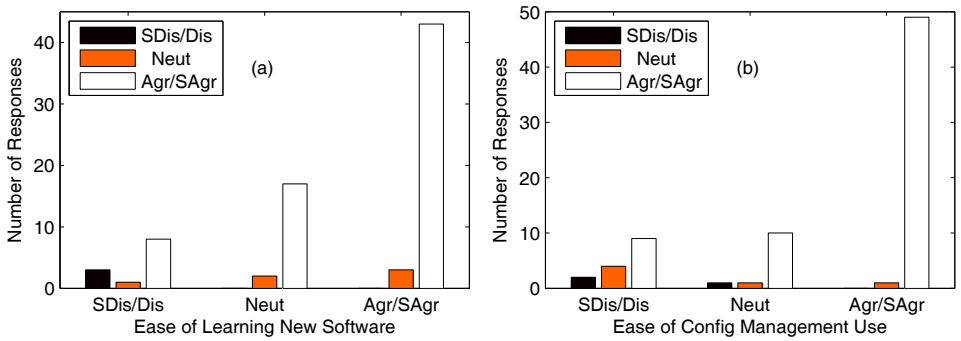

Figure 2: Student responses to (a) "I find it easy to learn new software"; and (b) "I found the Configuration Management Software easy to use". The categorisation is from the response in Figure 1.

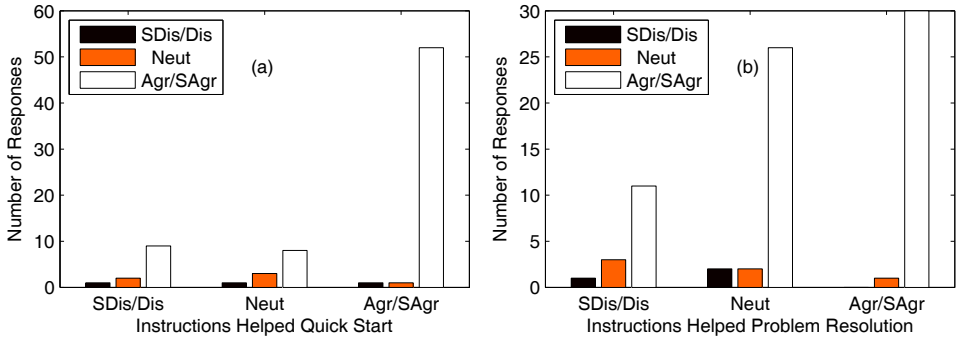

Figure 3: Student responses to (a) "I found the provided instructions helpful to get started quickly"; and (b) "I found the provided instructions helpful to resolve problems easily". The categorisation is from the response in Figure 1.

A set of instructions was provided to the students to help them get started and with some basic troubleshooting. These instructions appear to have been reasonably effective (**Error! Reference source not found.**), although they were less beneficial in terms of resolving problems. Students were given continuing support through a dedicated forum to resolve their difficulties, so it appears that this effort is necessary to account for the unforeseen problems that are inevitable.

> Respondent 24 (External, Strongly Agree, Agree): *The errors were diffuclt (sic) to understand and learn how to resolve, however the teaching staff on the forum were very proactive and speedy in their replies to help.*

Some of the capabilities that CM systems provide are identifying what changes have been made to files and who performed the changes. **Error! Reference source not found.** shows whether students like these capabilities and how they used them. Interestingly there was no negativity towards the ability to track file changes, so the students recognise the value in the capability even if they themselves found the system difficult to use and therefore did not use the system much and even if they did, only performed the most basic operations necessary: sharing the files.

Error! Reference source not found. shows that the vast majority of students accessed the system weekly, which is unsurprising given that marks were assigned (less than 5% of the total available marks) for the quality of the CM system's use. Therefore once some of the members became active in its use, the other members are forced to do so in order to access the team's project files. This can however lead to strategic behaviour from the students:

> Respondent 75 (Internal, At least once per week): *I will be happy if I never see this again, all that happened was one person submitted all files just to keep you happy*

Four students responded that they never used the CM software; interestingly all liked that the team had a central file repository. Respondent 10 noted great difficulty in getting the software to work because of having a Mac computer; this was the major difficulty encountered because the recommended software (TortoiseSVN) is only released for the Windows Operating System (OS). One of the students recommended using the SCPlugin software for students using an Apple OS, but this may have come too late for Respondent 10 to become engaged. Respondent 55 may have been confused as to what they were being asked at that point in the survey, because the comments all pertain to a separate tool used in the course for a different purpose. Correlated answers (e.g. "Not Applicable" to "I found the CM system easy to use") show that Respondent 71 genuinely did not use the system, but no comment was made to discover the reasons for this. Finally, similar correlated answers from Respondent 79 appear to support the statement of never using the software, but the comments show some understanding of its operation:

> Respondent 79 (Internal, Never): *(It is good that you can:) 1. Track who is making changes 2. Go back to older versions of files if newer one contains errors etc. (Where the system can be improved:) Cannot tell if someone is trying to make changes at the same time*

It must be concluded that while Respondent 79 did not use the software, some understanding of its use was gleaned by talking with peers.

One of the stated primary reasons for implementing the CM system was in order to simplify the process of staff assisting the students by having ready access to the complete suite of files that were being worked on. Some of the authors have taught and continue to teach in courses where this system was implemented, and it is our experience that it has improved our ability to respond to student enquiries quickly and directly, thereby removing waste communication. As a consequence, we wholeheartedly encourage the uptake of this tool. The students' perspective is somewhat mixed (Figure 6). While there were very few students who did not like this aspect, there was only a relatively small number of students who thought that this access improved their performance. A note of caution in the interpretation of this result is in order: the definition of "performance" may be interpreted differently, so that it is possible that it is taken as an indicator of grades, while another interpretation is that it helped the operation of the team in general. Further work will aim to distinguish between these potential interpretations. It is also possible that the effect is due to the quality of the facilitator for a team (Gibbings, Lidstone, & Bruce, 2010) and how they used the system.

> Respondent 28 (External, Agree, Agree): *The lecturer can run your code, or see your draft document, while you are talking to him and give instant feedback.*

Finally, students were asked about their perspectives on the usage of the system beyond the current course (**Error! Reference source not found.**). It seems somewhat surprising that there were so few negative responses to these questions. This is perhaps an example of students recognising some value in the system, so long as they do not have to use it themselves if they did not like it.

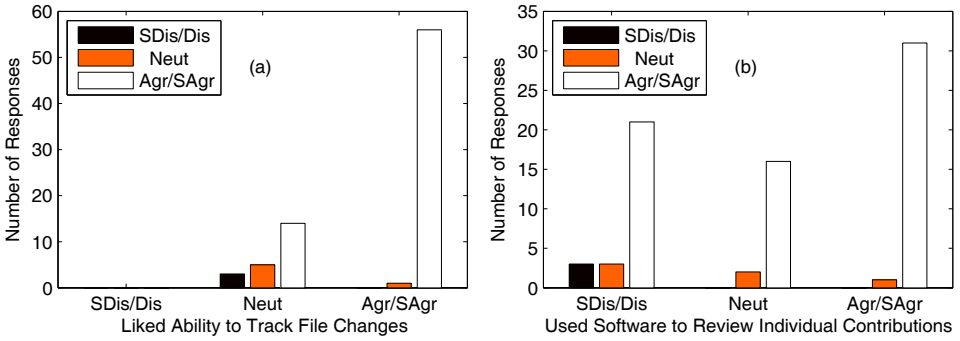

Figure 4: Student responses to (a) "I liked being able to keep track of all the changes to the files"; and (b) "I used the software to review individual contributions". The categorisation is from the response in Figure 1.

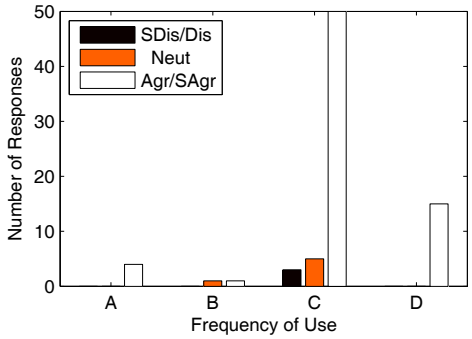

Figure 5: Student responses to "I used the configuration management software:" with the responses A: "Never", B: "At least once per month", C: "At least once per week", D: "At least once per day". The categorisation is from the response in Figure 1.

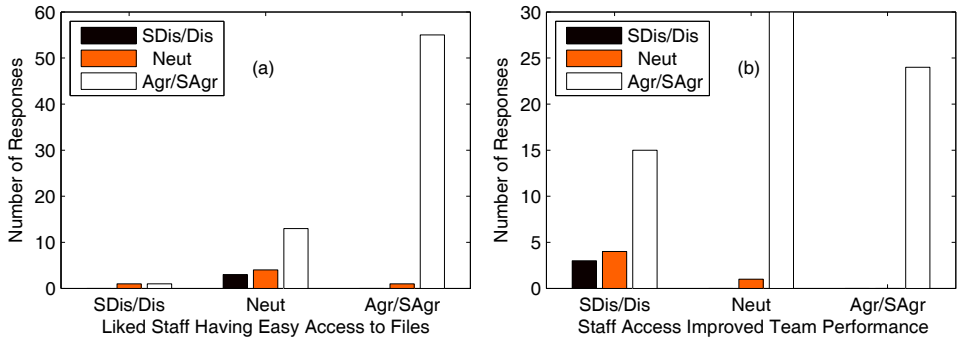

Figure 6: Student responses to (a) "I liked the staff having easy access to our repository"; and (b) "I thought that the staff having access to our repository improved our team's performance". The categorisation is from the response in Figure 1.

Conclusions

An investigation into the impact of configuration management systems on the student experience in a team project has shown that the majority of students found that it assisted them with completing their project. The primary objectives of allowing the students to store their common files in a rigorous manner and to control which version is active were largely achieved. Using configuration management tools also exposed students to best practice in file management widely used in industry.

The few students who did not like the implementation of the configuration management system in the course were found to be those who have difficulties with software packages in general and/or had difficulties in getting the configuration management software installed on their computer. This indicates that perhaps additional support is required to assist students who have difficulties with software.

There was some recognition by the students of the improvements that were afforded to them due to the ease with which the staff could access their files; the teaching staff found that the system made assisting students much easier.

Probably the best indicator of the impact this intervention has had is the take-up of the system in the subsequent course ENG4104, where it is not compulsory. Note that all members of the team would have to agree to use the system, so there may be some individuals who would have preferred to use it but the team decided not to. In 2012, there

were 59 students (out of 140: 42%) who used the system in 21 teams (out of 36: 58%) while in 2013, there were 38 students (out of 165: 23%) in 14 teams (out of 42: 33%). Because the rate of student usage was lower than team usage, this indicates that some individuals only downloaded files and this is not recorded in the usage statistics.

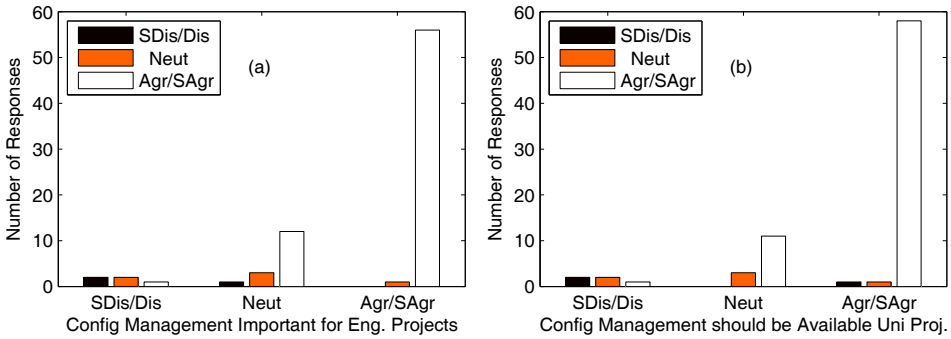

Figure 7: Student responses to (a) "I think configuration management is important for engineering projects"; and (b) "I think configuration management software should be available for use in team-based university projects". The categorisation is from the response in Figure 1.

Acknowledgements

The Engineering Education Research Group (EERG) at USQ provided financial support to analyse the data and attend the conference.

References

Broman, David, Sandahl, Kristian, & Baker, Mohamed Abu. (2012). The Company Approach to Software Engineering Project Courses. *IEEE Transactions on Education, 55*(4), 445–452.

Gibbings, Peter, Lidstone, John, & Bruce, Christine. (2010). How do student attributes influence the way students experience problem-based learning in virtual space? *Australasian Journal of Engineering Education, 16*(1), 69–80.

Hinsen, K., Laufer, K., & Thiruvathukal, G.K. (2009). Essential Tools: Version Control Systems. *Computing in Science & Engineering, 11*(6), 84–91.

IEEE. (2005). IEEE Standard for Software Configuration Management Plans (Vol. 828).

ISO. (2003). Quality management systems – Guidelines for configuration management (Vol. 10007). Geneva, Switzerland: International Organization for Standardization.

McDonough, Edward F., Kahn, Kenneth B., & Barczak, Gloria. (2001). An investigation of the use of global, virtual, and colocated new product development teams. *Journal of Product Innovation Management, 18*(2), 110-120. doi: 10.1111/1540-5885.1820110

NATO. (1991). NATO Material Configuration Management Policy and Procedures for Multinational Joint Projects (Vol. 4159).

Pinto, Mary Beth, Pinto, Jeffrey K., & Prescott, John E. (1993). Antecedents and Consequences of Project Team Cross-functional Cooperation. *Management Science, 39*(10), 1281-1297.

Reid, Karen L., & Wilson, Gregory V. (2005). *Learning by doing: introducing version control as a way to manage student assignments*. Paper presented at the Proceedings of the 36th SIGCSE technical symposium on Computer science education, St. Louis, Missouri, USA.

Spinellis, D. (2005). Version control systems. *Software, IEEE, 22*(5), 108–109.

USDoD. (1992). Military Standard – Configuration Management (Vol. 973).

Copyright statement