File 20110121.0310: Weekly activity report 0172:

weekly activity report 172 (loughry)
Joe Loughry
Sent: 21 January 2011 03:10
To: Niki Trigoni; Andrew Martin; Joanna Ashbourn
Cc: otaschner@aol.com; anniecruz13@gmail.com; andrea@hpwtdogmom.org;
chip.w.auten@lmco.com; edloughry@aol.com; diane@dldrncs.com;
Joe Loughry; mmcauliffesl@comcast.net; tom.a.marso@lmco.com

Weekly activity report no. 20110120.1308 (GMT-7) sequence no. 0172, week 1 HT

Julie Sheppard wrote to ask for a GSO.14b form for deferral of confirmation of status; I filled it out with a brief explanation of the reason for the delay and a copy of the proposed dissertation outline along with estimated dates for completion. The form needs to be signed by Dr Martin and by my college. Julie offered to get it around to where it needs to go for signatures; she also checked with MPLS and found that the GSO.14b form was all they really wanted. I sent it to Julie yesterday; the Fedex tracking number is 8685 8800 0700. By the time I am finished I will owe Julie some serious favours.

The Air Force sponsor asked for another emergency report today, this one on technical progress, project status and funding. It goes to a different organisation from last week’s emergency report; I will do it after I finish this report. My funding is reasonably secure through September on that research project, but I am preparing a FAFSA report in preparation for applying for another US federal student loan to cover travel expenses and fees through Trinity term. I plan to be in Oxford the week of 13th March with preliminary qualitative analysis (coding and categorisation) and a grounded theory for R’’ in hand for me to talk about. I am getting excited about grounded theory. It is clearly a good way to analyse data in software engineering projects where experiments are difficult or impossible to perform, either because such experiments would be impractical, or unethical, or too expensive to conduct. I really think this is a methodology I can use. Having started with the history of the method [Glaser and Strauss, 1967], I am concurrently making a list of codes and categories---deriving some of the structure of that list from the set of anonymisation codes already discovered---sufficient to code all of the case studies in the context of the abstract accreditor behaviour model while also being useful as event traces. I am looking forward to being able to use some artefacts from R’ that before being introduced to qualitative data analysis I thought I would not be able to use. I still do not have any results to report.

I obtained a draft copy of forthcoming government guidance titled ‘Best Practices Guide for Operating a Cross Domain Solution on a Virtualized Platform’ (NSA Information Assurance Directorate (IAD), dated 19th January 2011). It relates to work that the case study developer (see below) is doing. I also encountered a reference earlier in the week to a paper by Flchais and Sasse (2009) that argued for the idea that assignment of liability is what motivates stakeholders; this is relevant to the actions of accreditors, I might argue, in CDS C&A. I think this will tie into all three grounded theories as well as the risk market interpretation in Chapter 7.

Training I attended this week included two webinars, one on DO-178 aircraft safety-critical code certification conducted by Adacore (with an interesting comparison between DO-178 Level A to the requirements of Common Criteria EAL4) and the other presentation on systems engineering thinking. I missed Dr Martin’s seminar about inter-disciplinary research in Information Security on Monday due to technical problems with the audio-visual link to the room it was being held in, but I had already read the background paper and I will be attending next week's seminar by Colin Williams. Lockheed work---or more specifically the Air Force research contract that provides most of my funding---has been occupying more than its share of time again. I have got to try harder to limit it to no more than 20 hours per week, allocating twice again that amount of time to thesis research. I have not got the project planning software configured yet; I have documentation that explains how to set it up but writing two reports for the Air Force took up all my time this week aside from reading about grounded theory. I will get a plot showing the fraction of each task on a 168-hour week time-line drawn before the end of the weekend.

I want to look up the following papers on Value Scenarios [Nathan, 2007]; diffusion of responsibility [Darley, 1970]; and risk management [two papers: Flchais, 2009; and Flchais and Sasse, 2009].

Security Reading Group met Wednesday to discuss Shamal's draft of 'Bringing mis-usability home: finding and resolving mis-usability with Mis-usability Cases' to be submitted to the BCS HCI conference. I was not the only person in Reading Group who immediately brought up the same concern with this paper: that it ignores an important case in secure software engineering---the not uncommon case in which all users of the system should be treated as potential attackers. Shamal argues---and this paper is intended for an HCI conference, so the the threat argument is turned down a bit for the audience---that it is harmful to the analysis to consider users to be attackers. There is an obvious connection to the paper by Adams & Sasse (1999) here, but also to Whitten & Tygar (2002). Three of the participants in Reading Group counter-attacked on that one point. We compromised by asking Shamal to make clear in the introduction of the paper that it applies only to that subset of systems where not all users are potential attackers.

In the view of misuse cases [Alexander (2003) et seq.], the user is always a potential attacker. The author suggests in this paper that the foregoing in ibid. comprises an unnecessarily limiting analysis constraint. HCI people, he argues, do not think in that way. This led to a discussion of alternative names for the new concept: 'abuse cases', 'anusability cases', and 'unsecusability' were suggested. None of them seemed to communicate the essential core of the idea as well as the ungainly and awkward but parse-able'mis-usability' so we decided to accept it for now.

The Therac-25 was brought up as a slightly misaligned parallel example. It is bad when a bad design fails, but worse when a bad design turns a non-malicious user into a potential attacker. The system creates the damage to itself. In Section 4.3, misuse cases [ibid.] can be anything at all. Mis-usability cases, on the other hand, must satisfy a Use Case; that is the distinction Shamal is trying to draw in the paper. All preconditions and postconditions of the associated use case must be satisfied before a mis-usability case is considered valid. Several people in Reading Group had problems with that, but I think it is a useful refinement. 'Obstacles' are ways that a 'Requirement' can be 'Obstructed' in this formulation. I wish the paper were longer; it would benefit from a couple of major additional sections explicitly laying out the constructed terminology with formal definitions and at least one example of each. Shamal acknowledged this, saying that the page count was extremely limited by the conference and that in fact the editors had lowered the limit by one page yesterday. He told us about a pair of sections he had taken out, and I pointed out that one of the removed sections was clearly implied by the remaining narrative in Sections 4 and 5, so he got that concept for free. I thought it was clearly implied by the text.

John asked whether mis-usability cases have an application in non-security contexts. In the book Design Noir [Dunne, et al. (2001)] the authors argued that any human system necessarily has unintended consequences. The design noir people would say that mis-usability can never be removed completely from any real system. Shamal is trying to extend that thought into a software engineering methodology.

There is no reason why Mis-usability Cases and Misuse Cases could not be used at the same time. I would put it as saying their moving parts do not conflict. This mis-usability cases technique is interesting because it can be applied late in the process. Maybe, suggested Shamal, thinking about security up front is not the right approach after all. Doing it at the end spells disaster; everyone agrees with that. Doing it as soon as possible, he argues, is better than either not doing it at all or doing it too late. Everyone concurred that the major contribution of this paper, unlike what was claimed at the end---Shamal is going to rewrite that part---is really the advancement of a method that can be applied practicably at a rather late point in the design process. That is the new thing here. The author demonstrated successfully in his case study that even with the very realistic and familiar stumbling blocks to the application of the methodology described in Section 4, the method nevertheless was successfully applied and successfully discovered actual, significant requirements at a surprisingly late stage (by the standards of software engineering methodologies, and especially for security engineering methodologies) in the software development life cycle of the project.

At the end of the meeting we contributed a full report on the many typographical, verb tense agreement, citation style, figure clarity and cut-and-paste errors in the document. Shamal promised to fix them before submitting.

Regarding case study R'', the RM software developer was merged this week with a smaller CDS programme in Lockheed Martin. The new programme manager is the old programme manager of the smaller programme; he is now in charge of the combined programme and the current RM programme manager will become his deputy. Trusted Manager (TMAN) is a much smaller programme than RM---fewer than a hundred installations for approximately five customers vs more than 400 sites and perhaps 100 active customers in the case of RM. The programmes are at comparable levels of historical development. The new programme manager stated his intention in an all-hands meeting to merge the two CDS products as well as the development organisations. A third CDS programme---presently called Next Generation but which does not currently have a product associated with it---for the next six months will exist nominally between RM and TMAN as the new programme manager tries to consolidate the organisation and merge the technical capabilities of both products and both development/support organisations into one. RM software developer personnel appear to be nervous about the merger, probably because of the new programme manager's history and background. The two CDS alternatives have long been competitors in some of the same markets although their technical competencies and especially their government certifications for handling classified information do not completely overlap. The new programme manager had some interesting things to say about the preference of TMAN customers for government certification; TSABI data owners, in that instance, seem to care less about the results of regression testing and IV&V at the government level than has been the case with RM customers in the SABI world. The consolidation effort begins immediately with easily foreseen cost savings such as elimination of duplicated IAVA tracking. In future, the new programme manager would like to actually merge the two products into a service to be offered to customers and to offer it across a range of sizes from enterprise to small form factor. In another meeting, it was explained differently: that there would be a suite of products including RM, TMAN and maybe one other, with the intent of being able to satisfy any CDS requirement.

TMAN has some structural cost advantages over RM: the TMAN programme is unclassified and has an unclassified software baseline. Their overhead for physical and information security is accordingly smaller. Some comparative numbers were mentioned (not reproduced in this report) suggesting to me that the relative cost factors of development in the two programmes are different---and in some ways the same---in interesting ways. Both programmes recently ported their software baseline from Trusted Solaris 8 to Solaris 10. (In each instance, third-party vendor hardware life cycle support realities provided the necessary impetus for a change that developers of certified software will never incur willingly.) The smaller programme (N < 100) spent almost exactly 80 percent of the larger programme's (N > 400) cost for a similar amount of development effort but with approximately 1/3 the recertification testing, estimating from the current difference between TSABI and SABI levels of effort. Given comparable levels of efficiency, I would have expected closer to 50 percent.

Independently, the status of R'' post-certification activities (and waiting for official SABI approval) is proceeding normally. Version 5.01 (currently at build 5.01c) is one week away from the final CSCI. 5.01 will have a full-blown FAT; IV&V personnel plan to travel to the developer site to witness the acceptance tests. The release contains lots small bug fixes, the sort of things that could not have been discovered before 5.0 went into production, and aligned with CT&E expectations. Version 5.02 development will be started immediately after FAT. 5.02 is expected to be an enhancement release, and will not be ready for acceptance testing until Autumn, about the same time that 5.01 is certified according to the schedule. Some of the enhancements in version 5.02 are being done to reverse the effect of design decisions made at the start of version 5 development, when the port to a significantly different new operating system (for the aforementioned third party hardware vendor product cycle reasons) prompted adoption of certain software technologies that have since proved either technically suboptimal, unnecessarily complicated, or suddenly more expensive as a result of unrelated OS vendor pricing changes.

My current tasks, in priority order, are:

0. Finish emergency report for funding sponsor [must be done by tomorrow morning].

1. Continue reading on grounded theory history and coding methodology.

2. Refine codes and categories for R'' that will also work for R' and R-zero. Keep them at an appropriate level of granularity to still be useful for event traces.

3. Figure out if I can use the chronological record in my lab notebook

as a source for the ‘memo writing’ activity that occurs later [waiting].

4. Plot tasks on a new Google Calendar as blocks in a 168-hour week. Establish limits on non-thesis work times [Sunday].

5. Outline the survey journal article that the assessors asked for. I still need to find some good examples of how to write a survey article.

6. Outline new Chapter 1, revised Chapter 3, and completely new Chapter 4 [want to have these ready to take to Oxford in March]

7. Buy plane tickets and finish student loan application.

Joe Loughry
Doctoral student in the Computing Laboratory,
St Cross College, Oxford

End of WAR 0172.

References