Journal of Applied Computing and Information Technology
Volume 23, Issue 1, 2019 --- ISSN 2230-4398

The Avocado Pincer– Towards Giving Git/GitHUB a Better Squeeze

Todd Cochrane
NMIT, Nelson, NZ
Todd.Cochrane@nmit.ac.nz

Lars Dam
NMIT, Nelson, NZ
Lars.Dam@nmit.ac.nz

Ali Kahwaji
NMIT, Nelson, NZ
Ali.Kahwaji@nmit.ac.nz

Neil Benson
NMIT, Nelson, NZ
Neil.Benson@nmit.ac.nz

ABSTRACT

In this paper, the first design of a novel approach to determine the application of new technology in our courses is described. How to determine current teaching and learning as well as student situations in our courses is part of the first phase of this design research centred project. The "avocado pincer" that focuses on student, tutor and "new IT Tech" concerns has become the guiding frame. As part of this design process, we have also discovered a useful encoding process for course delivery and assessment that allows comparison of courses in their delivery and assessment patterns, which could lead further reflection on overall delivery of programmes. At this point we are working on categorisation of student concerns, and the determination of suitable affordances in the case of Git/GitHUB. The design of an "avocado pincer", successfully enhances in our decision processes.

Keywords: Educational Technology evaluation, Git/GitHUB, design based research, new IT Tech

1. INTRODUCTION

We posit as Information Technology (IT) tutors at New Zealand polytechnics, in response to a dynamic ICT landscape (Lahey, 2005), we continuously integrate new IT into our courses. Hence, we need a way to assess the utility of new technology in teaching and learning. On-going integration of "new IT Tech" into our courses (as with courses from all disciplines) can be seen from the tutor perspectives and the student perspectives. According to Henderson, Selwyn, and Aston (2017, 1578) in students' perception of "what works" the value of new tech tends towards logistical utility, despite educators' perception of new IT Tech as an enhancer of learning.

New IT Tech like the trendy avocado, needs to be squeezed without damaging it before you buy it. In in this paper we present the on-going design of a newly invented "avocado pincer", for evaluation of new IT Tech. Use of Git in Computing courses has been documented and assessed (Glassey, 2019, Griffin & Seals, 2013, Gunnarsson, Larsson, Månsson, Mårtensson, & Sönnerup, 2018). In this case our target "new IT Tech" is Git/GitHub.

2. DESIGING A METHOD

Design research is applied in the design of the "avocado pincer". According to Reeves (2000) , design research has four phases, each of which generates knowledge: 1) "Analysis of Practical Problems by Researchers and Practitioners", 2) "Development of Solutions with a Theoretical Framework", 3) "Evaluation and Testing of Solutions in Practice", and 4) "Documentation and Reflection to Produce 'Design Principles' " (Reeves 2001, Figure 3 p. 9). This paper reports on phase one activities, in which analysis towards evaluation of our target "new Tech" Git/GitHUB has led to the discovery of the pincer.

Depicted in Figure 1, student and tutor perspectives are thought of as two corners of a triangular pincer that is applied to determine suitable evaluation of new technology placed at the top of the pincer. Tutors are concerned with advantages in their teaching practice as well as enhancing learning. From their perspective the "new Tech" should allow some pedagogic gain as well as aligning with students becoming vocational practitioners, i.e. members of their chosen discipline. Students see gain in logistic use in their day to day learning.

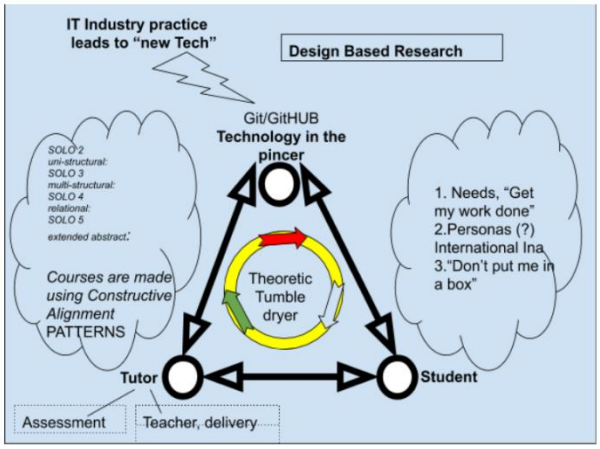

Figure 1. The "avocado pincer". Tutor perspectives, with Student perspectives in learning and teaching dynamic are considered in the context of suited theoretical frames; the Theoretic Tumble dryer.

The students are also concerned with alignment of their study with practices in the discipline they are joining.

At our institute constructive alignment (CA) (Biggs, 1996) is actively applied in the design and delivery of courses, the on-going learning is aligned to the assessment and the assessment is aligned with learning outcomes. Learning outcomes in turn align with industry requirements. Design of the pincer "Git/GitHUB" hence required an overview of the way our courses would align with Git/GitHUB in the context of CA as it has been applied in our courses, and an overview of our students' perspectives in the application of Git/GitHUB in their learning trajectories.

In the case of the tutor requirements, courses were considered in terms of their delivery and their assessment patterns. Hence from those patterns an inference of the effect on the programme

can be inferred. Forty courses were identified from the Bachelor of Information Technology (BIT) and Computer Graphics Imagery (CGI) programmes. Course outlines and assessment details of each of those courses were gathered and an encoding system for assessment and delivery patterns was derived.

In both cases categories were identified based on the types of assessments and delivery styles. Table 1 lists and describes five categories of assessments identified through analysis of assessments from the programmes.

Table 1: Categories of Assessment

Category	Description
Spot Check Assignment (SCA)	The assignment provides tutors and students with formative and summative assessment towards learning outcomes – these are like classical assignments in didactic teaching and learning with a series of problems and short answer questions.
Spot Check Test (SCT)	The test provides tutors and students with formative and summative assessment towards learning outcomes – these are like classical assignments in didactic teaching and learning with a series of problems and short answer questions.
Project (P)	An on-going assessment, based on a specific product that is to be delivered by the student, formative and summative feedback is provided through out and in the final delivery of developed product – these assessments are also used as a vessel that drives delivery of course content.
Laboratory work (L)	The student attends a series of laboratories in which they cover specific learning outcomes – feedback is usually summative, and incremental, the "labs" cover small chunks of learning outcomes.
Learner Journal (LJ)	The student writes a reflective journal on the coursework – details of requirements vary from course to course, for example some cases the journal is a record of reflection on laboratory work, in others on project work, and in others the Learner Journal meets specific learning outcomes.

Assessment patterns were then encoded for level five courses and some level seven courses, using the encoding scheme shown in Table 2.

Table 2: Assessment encoding

Assessment Encoding is a sequence of five comma separated annotations
For example, A1,1,16,P,I
Code 1. The first code represents the assessment number in the course, i.e. An (where n is the assessment number in the course)
Codes 2 and 3. The following two codes indicate the week in the course the assessment was started , followed by the week the assessment finished
Codes 4 and 5. The last two are assessment type – following by G or I. **G** indicates a group work based assessment, **I** indicates an individual assessment.

The encoding of the courses identified courses with similar assessment patterns, and hence potentially similar affordances that could be provided by Git/GitHUB. For example, encoding identified that all level 5 CGI courses, and the BIT course CSA501 and NET501 have the same assessment pattern, whereas the BIT course TEC501 has potentially different requirements by presenting a different assessment pattern.

As shown in Table 3, five categories of delivery style were identified in our courses.

Table 3: Categories of Delivery style

Category	Description
Strict didactic (L)	This delivery style is the delivery where the tutors presents learning in a classical style. Leaning is cyclic, with students encouraged to ask questions and practice the concepts.
Problem based (P)	The tutors and students work together on specific problems – which unfolds learning trajectories. Running a project is an example of problem-based delivery.
Topic/Lab/Skill Development (T)	Focus on a specific skill, the students gain a skill required by their discipline in practical situations – practice centred learning and delivery.
Critique (C)	In which the learner presents material that indicates their achievement and tutors provide feedback to the learners that constructively critiques their practice and reflections. This is applied to journals, and at milestones hand ins, and in project final delivery by the students.

The encoding of the delivery styles identified similarities between courses. Courses delivered by different tutors tended to have similar delivery styles. Not all courses used the same delivery style throughout, for example there is a pattern where early in the courses there is more L style delivery, with P being used towards the end of the course. These courses are aligned with their upcoming assessments. There is a notion that the pattern where skills are established then used in a practical context is potentially useful in our vocational contexts.

Table 4: Delivery style encoding

Delivery Encoding is a sequence of four comma separated annotations
For example, D1,1,16,L
Code 1. The first code represents the delivery number in the course, i.e. Dn (where n is the sequence of the delivery style in the course)
Codes 2 and 3. The following two codes indicate the week in the course the delivery style was started, followed by the week it stopped.
Codes 4. The last is the delivery style code.

An example of mixed delivery styles is the SDV701, that is encoded "A1,5,5,SCT,I" , "A2,8,16,P,I" with "D1,1,4,L" followed by "D2,5,16,P" , that shows this change in delivery style to meet the assessment mode.

3. WHAT ABOUT STUDENT REQUIREMENTS?

Current work is heading towards developing the students' perspective, see Figure 1. The current work is leading to tabulation of affordances provided by Git/GitHUB in context of both tutor and student perspectives. Student perspectives provide the other arm of the pincer. Two approaches to categorisation that can be applied to determine student requirements have been considered, a student persona inventory, and the application of the SOLO taxonomy.

Student personas (Smith et al., 2016) have been considered as a potential indicator of student requirements in the selection of technology and use in our courses. They are concerned with

identifying matches in requisite competencies. Based on an approach of identifying personae (Yström, Peterson, Sydow, & Malmqvist, 2010), research extends on categories personas to into a matrix of competences. The selection of competences in the article presents appears relevant, but a further justification is hard to find. Yström et al. (2010) clearly link their choice of relevant competencies back to the learning outcomes in their curriculum, when personae categories are generated. This perspective can be used if we create our own, justified categories, based on justified competencies that we link back to in our learning outcomes.

Using the SOLO taxonomy and constructive alignment (Biggs, 1987, 1996, 2011) as a filter to determine student centred requirements. Sitting in the same domain as Blooms taxonomy that has three levels of increased complexity,SOLO has five levels (where we want to reach the four top levels) of increased reflection with the following student processes: Explain, Relate, Prove and Apply. Biggs (1987, 1996, 2011) combines these with the theory of constructive alignment, stating that the teachers' intention, must be aligned with the assessment structure to make the student's learning activity supported by the both the teachers' intention and the assessment. The alignment must/can happen following the SOLO taxonomy.

The motivation for presenting them in this context is that Biggs (1987, 1996, and 2011) has this interesting split between teaching activity and assessment that we made in our course analysis. Hence, we are taking this alignment for granted - to a large extent. Subsequently, have stated this as one factor in our other triangular model depicted in Figure 1. What we/the institution does - what the student does - and what technology we are applying. Another motivation is that Biggs' perspective creates three levels of teaching styles (from lowest to highest):

- What students are?
- What teachers do?
- What students do?

Where we want to be in this picture, is that we are concerned with what students do, but need to understand where they are in terms of competencies to be able to meet their demands and design learning activities building on where there are. In other words, a non-judgement design process.

4. WHAT ABOUT GIT/GITHUB?

Version control systems like Git combined with hosting services like GitHub are playing a vital role in the education of universities and institutes students. With these both tools have evolved to serve the educational sector needs. Tutors in both spaces of class locally and remotely can utilize and control a number of these technologies to distribute course materials and engage with their students, including tools such as Learning Management Systems(LMS) that are focused on education, other tools that can be created for different used in an educational context like social media platforms.

The literature indicates research into Git/GitHUB as a tool in classroom practices that can be usefully applied to determine important attributes that suit educational contexts (Deursen, 2013; Feliciano, Storey, & Zagalsky, 2016; Gennarelli, 2017; Griffin & Seals, 2013; Gunnarsson, Larsson, Månsson, Mårtensson, & Sönnerup, 2018; Hsing & Gennarelli, 2019; Kelleher, 2014; Lawrance, Jung, & Wiseman, 2013; Lupo, 2017; Matthies, Treffer, & Uflacker, 2017; Miyashita, Hazeyama, Hashiura, Goto, & Hirasawa, 2018; Vasilescu, Filkov, & Serebrenik, 2013; Vihavainen, Vikberg, Luukkainen, & Pärtel, 2013; Zagalsky, 2017). More specifically, it can be seen that Bishop, Jensen, Scacchi, and Smith (2016) include Git/GitHUB in their research into how to use OpenSource software in classrooms, Bonakdarian (2017) descibes the application of Git/GitHUB in undergraduate Computer Science classes, Cajander et al. (2012) describes collaboration between students in an international project using Git/GitHUB tecnology.

Git/GitHUB is a key factor tool to manage asynchronous and distributed software workflow, also it is an important component of managing resources and teamwork in any software development environment. This has led to a demand upon tutors and to introduce Git to software development students and to preparing them for their future work environment and that by including version control systems in the courses. Git will help students with their software projects by organizing their workflow, teamwork and assign any specific task to any member in team. At this point the "avocado pincer" leads the review of Git and GitHub as applied to classroom and educational contexts.

5. SUMMARY

Designing a way to evaluate "new Tech" before we introduce the technology into our classrooms has led to the design of a novel approach evaluating the Tech, before it is integrated into our courses. The "avocado pincer" goes beyond a cost benefit analysis, extending evaluation into a design research process that attempts to integrate a wide range of relevant concerns.

Evaluating the IT Tech to see where its affordances enhance both our delivery and assessment and where it leads to alignment for both pedagogic as well as vocational content knowledge (Shulman, 1986; Mishra & Koehler 2006) increases our ability to determine how that IT Tech can be used and applied in ITP programmes.

6. REFERENCES

Biggs, J. (1996). Enhancing teaching through constructive alignment. Higher Education, 32(3), 347–364.

Biggs, J. B. (2011). Teaching for quality learning at university: What the student does. McGraw-hill education (UK).

Biggs, J. B. (1996). Enhancing teaching through constructive alignment. Higher Education, 32(3), 347–364.

Biggs, J. B. (1987). Student Approaches to Learning and Studying. Research Monograph. ERIC.

Bishop, J., Jensen, C., Scacchi, W., & Smith, A. (2016). How to use open source software in education. Proceedings of the 47th ACM Technical Symposium on Computing Science Education, 321–322. ACM.

Bonakdarian, E. (2017). Pushing Git & GitHub in undergraduate computer science classes. Journal of Computing Sciences in Colleges, 32(3), 119–125.

Cajander, Å., Daniels, M., Cullhed, M., Clear, T., McDermott, R., & Laxer, C. (2012). Categorizing how students use Collaborative Technologies in a globally distributed project. 2012 Frontiers in Education Conference Proceedings, 1–6. IEEE. Deursen, A. van. (2013, December 30). Teaching Software Architecture: With GitHub! Retrieved August 29, 2019, from Arie van Deursen website: https://avandeursen.com/2013/12/30/teaching-software-architecture-with-github/

Feliciano, J., Storey, M.-A., & Zagalsky, A. (2016). Student experiences using GitHub in software engineering courses: A case study. Proceedings of the 38th International Conference on Software Engineering Companion, 422–431. ACM.

Gennarelli, V. (2017, March 2). Real-time feedback for students using continuous integration tools. Retrieved August 29, 2019, from The GitHub Blog website: https://github.blog/2017-03-01-real-time-feedback-for-students-using-continuous-integration-tools/

GitHub. (2019). Build software better, together. Retrieved September 10, 2019, from GitHub website: https://github.com

Glassey, R. (2019). Adopting Git/Github within Teaching: A Survey of Tool Support. Proceedings of the ACM Conference on Global Computing Education, 143–149. ACM.

Griffin, T., & Seals, S. (2013). Github in the classroom: Not just for group projects. Journal of Computing Sciences in Colleges, 28(4), 74–74.

Gunnarsson, S., Larsson, P., Månsson, S., Mårtensson, E., & Sönnerup, J. (2018). Engaging Students using GitHub as a Learning Management System. Lunds Universitets Pedagogiska Utvecklingskonferens 2017, 63–70. AHU, Lunds universitet.

Hsing, C., & Gennarelli, V. (2019). Using GitHub in the Classroom Predicts Student Learning Outcomes and Classroom Experiences: Findings from a Survey of Students and Teachers. Proceedings of the 50th ACM Technical Symposium on Computer Science Education, 672–678. ACM.

Kelleher, J. (2014). Employing git in the classroom. 2014 World Congress on Computer Applications and Information Systems (WCCAIS), 1–4. IEEE.

Lawrance, J., Jung, S., & Wiseman, C. (2013). Git on the cloud in the classroom. Proceeding of the 44th ACM Technical Symposium on Computer Science Education, 639–644. ACM.

Lupo, C. (2017, July 11). Learn by doing at Cal Poly with GitHub and Raspberry Pi. Retrieved August 29, 2019, from The GitHub Blog website: https://github.blog/2017-07-11-learn-by-doing-at-cal-poly-with-github-and-raspberry-pi/

Matthies, C., Treffer, A., & Uflacker, M. (2017). Prof. CI: Employing continuous integration services and Github workflows to teach test-driven development. 2017 IEEE Frontiers in Education Conference (FIE), 1–8. IEEE.

Mishra, P., & Koehler, M. (2006). Technological pedagogical content knowledge: A framework for teacher knowledge. The Teachers College Record, 108(6), 1017–1054.

Miyashita, Y., Hazeyama, A., Hashiura, H., Goto, M., & Hirasawa, S. (2018). A Visualization System of the Contribution of Learners in Software Development PBL Using GitHub. 2018 25th Asia-Pacific Software Engineering Conference (APSEC), 695–696. IEEE.

Reeves, T. C. (2000). Enhancing the worth of instructional technology research through “design experiments” and other development research strategies. International Perspectives on Instructional Technology Research for the 21st Century, 27, 1–15.

Smith, H., Smith, L., Trounsen, R., Flannagan, T., McCarthy, D., & Sutton, K. (2016). Use of personas in education - A case study of development of core competencies. CITRENZ 2016, 11.

Shulman, L. S. (1986). Those Who Understand: Knowledge Growth in Teaching. Educational Researcher, 15(2), 4. https://doi.org/10.2307/1175860Yström, A., Peterson, L., Sydow, B. von, & Malmqvist, J. (2010). Using Personas to Guide Education Needs Analysis and Program Design. Proceedings of 6th International CDIO Conference, Montreal, Canada.

Vasilescu, B., Filkov, V., & Serebrenik, A. (2013). Stackoverflow and github: Associations between software development and crowdsourced knowledge. 2013 International Conference on Social Computing, 188–195. IEEE.

Vihavainen, A., Vikberg, T., Luukkainen, M., & Pärtel, M. (2013). Scaffolding students’ learning using test my code. Proceedings of the 18th ACM Conference on Innovation and Technology in Computer Science Education, 117–122. ACM.

Yström, A., Peterson, L., Sydow, B. von, & Malmqvist, J. (2010). Using Personas to Guide Education Needs Analysis and Program Design. Proceedings of 6th International CDIO Conference, Montreal, Canada.

Zagalsky, A. (2017, April 5). GitHub Issues and user testing as authentic assessment. Retrieved August 29, 2019, from The GitHub Blog website: https://github.blog/2017-04-05-github-issues-and-user-testing-as-authentic-assessment/