**Purpose:**

The PEAC meeting was held on 3rd May 2016 with a primary focus on discussion of planned changes to the BIT curriculum consequent upon the updating of a number of papers in 2016. The main purpose was to solicit input from the industry representatives around those modifications.

**Attending:**

Rob Broadley EAD

Paul Campbell Moonbase

Tom Clark BIT

Joy Gasson BIT

Patricia Haden BIT

Chris Lee Canonical

Darren Murray Independent IT consultant

Dale Parsons BIT

Thomi Richards Canonical

David Rozado BIT

Hamish Smith CIT

Lesley Smith EAD

Rebecca Twemlow Firebrand

**Apologies:**

Mark Oliver MTF

Ian Simpson Logic Studio

**Discussion:**

Prior to the meeting, Tom Clark had distributed to attendees a document outlining the intended curricular changes and specifying several questions for the attendees to consider.

*Changes:*

|  |  |
| --- | --- |
| PC Maintenance | Broadened to include a wider variety of devices and platforms. |
| Introduction to Systems Analysis | Object-oriented modelling material removed; replaced with relational modelling. |
| PP1 and PP2 | Heavily reworked to be IT-focused and work-integrated. |
| Games Programming in C++ | Extended to include a range of programming paradigms. |
| Networks 3 | Updated/modified for greater industry focus. |
| 3rd Year Project | Significant changes to structure, implementation, content and pedagogy to align to current industry practices. |
| Software Engineering | " |

*Questions:*

1. Are the overall changes to the programme reasonable?
2. Are there any other current papers that should be changed, replaced or removed from the degree.
3. (Regarding the marking of 3rd year project) How are you measuring developer and team performance in your organisations? Is it working well?
4. Are these performance measure suitable for project student assessment.
5. What are the core software engineering tools and methods that should be covered in [the software engineering paper]?
6. When you bring a new developer into your organisations, how do you introduce them to team practices like SCM use, coding style, etc.?
7. All students are required to take software engineering, even though they may not all plan to become software engineers. Is this position reasonable? Is it possible that other topics should be offered as alternatives?

*Feedback:*

1. **Are the overall changes to the programme reasonable?**
   1. More detail was requested regarding the planned changes to C++ Games (now called Programming 4). PH explained proposed new structure. Attendees were strongly in favour of including a functional programming language in the survey portion of the paper, and recommended a number of specific paradigms/topics/ languages: Go, ANSII C, pointers, memory management, multi-threading, automated testing. (*TR: "I hire every Go programmer I can find.")*
   2. RT asked where the non-technical content (*RT: "Communication, research, faking it")* would be covered in the updated curriculum, noting that historically OP BIT graduates are strong in these areas, making them desirable employees. TC gave further detail about PP1 and PP2 explaining the general shift in those papers from communications theory to communications practice applied in an IT context. JG noted that communication and people skills are considered essential and are embedded in all papers through, for example, frequent group projects, overt discussion of professional ethics, etc.
   3. TR asked how databases were being taught. TC described structure of Systems Analysis (name possibly to be changed to Databases 1) and DB2 and discussed currently fluid role of DB3. PH noted that databases were also used in most of the 3rd year programming and ops thread papers as a natural part of application development and systems management. TR noted that databases were best taught in a realistic context.
   4. CL asked about NoSQL, and stated that basic exposure to the topic (as currently provided in DB2) was sufficient.
   5. DR asked to describe Machine Learning paper (currently being offered in place of DB3). Extremely positive response.
   6. TR advocated for more comprehensive teaching of testing, while noting that it can be difficult to motivate strict testing protocols when developing new applications. Rather, the compelling case for testing is seen when changing legacy code upon which people are already depending, where proper unit testing gives assurance that existing functionality is not compromised. CL mentioned the motivational value of fear of existing users getting angry at you.
   7. TR asked for detail about the path for students interested in an operational role. TC described planned updates to Linux (transformed to Systems Administration 1) and current work-integrated model of the advanced Systems Administration paper. TR approved general approach, but suggested the need for greater emphasis on testing and real deployment and identified 3rd year project as a potential area for this change. TC described the "student-maintained BIT infrastructure" project concept as an opportunity for exposing students to realistic testing, deployment and maintenance requirements.
2. **Are there any other current papers that should be changed, replaced or removed from the degree.**
   1. Felt to be subsumed by preceding discussion. No specific suggestions for addition or omission of papers, but TR suggested the inclusion of architecture for massive scale.
3. **(Regarding the marking of 3rd year project) How are you measuring developer and team performance in your organisations? Is it working well?**
4. **Are these performance measure suitable for project student assessment?**
   1. TC expressed concerns about achieving an ecologically valid marking protocol for 3rd year project. Attendees were asked how they evaluate -- and are evaluated -- on the job.
   2. TR described strict code review and 360 feedback process. Noted that this approach is especially effective as identifying systemic personal weaknesses, and allowing them to be addressed.
   3. CL and TR noted that the issues identified in 360 tend to be more personal (e.g. not being able to take criticism gracefully) than technical (i.e. not being able to code). [*Ed: both CL and TR are at Canonical, which has a very stringent technical evaluation process for hiring. At a student level, code review and 360 should also identify skills and knowledge weaknesses.*]
   4. RT uses strict code review and evaluates her employees' ability to accept criticism. She also recommended a 360 approach, especially incorporating peer feedback.
   5. CL suggested incorporation of participation and engagement in team processes, but noted that it will be difficult to quantify this for marking.
   6. TR suggested that marking might be viewed as similar to dealing with an underperforming employee -- evidence of the problem(s) should be collected, and criteria for improvement established.
   7. PC stated that individual contribution must be incorporated into any marking protocol (*PC: "There's often that one student who is carrying the others.")*
   8. DM asked for more details about team composition. TC noted that all team members are expected to perform all major tasks during the course of their project (i.e. students can't only write code, or only do documentation, etc.).
   9. PC wanted marking to encourage innovative thinking*,* hoping to see more start-ups in Dunedin.
   10. TR queried how the requirement to engage with Open Source was being managed. TC explained that it was via extensive use of OSS for the implementation of projects, providing opportunities for contribution of patches, bug fixes, etc. Consensus among attendees of the importance of this activity for modern IT grads for strengthening the CV, building relationships in the industry, etc. TR notes a steep learning curve for the process.
5. **What are the core software engineering tools and methods that should be covered in [the software engineering paper]?**
   1. TC describes the relationship between the SE paper and Project as: SE covers what you need to know; Project lets you use it.
   2. Positive consensus on the specified indicative content for the new Software Engineering paper, particularly the breadth (see TC's original memo, the updated course descriptor, and further discussion below).
6. **When you bring a new developer into your organisations, how do you introduce them to team practices like SCM use, coding style, etc.?**
   1. PC has no company-specific coding style, preferring employees who are flexible enough to adjust to the style of an existing/legacy code base.
   2. PC takes a similar position about SCM, suggesting that one not over-emphasise a specific tool but ensure understanding of the underlying principles, allowing graduates to adapt to whatever work environment they find themselves in. CL concurs. (*CL: "Graduates need to know branching, not just know Git.")*
   3. RT involves the whole team (n=11) in hiring decisions and provides cross-over between the departing employee and the replacement, but has no specific on-boarding process.
   4. DM notes the importance of referral.
7. **All students are required to take software engineering, even though they may not all plan to become software engineers. Is this position reasonable? Is it possible that other topics should be offered as alternatives?**
   1. DM suggested this was a non-issue as the BIT as outlined was only producing software developers. PH and DP described the networking, ops and hardware papers that enabled students to prepare for careers other than straight development. TC noted that the current meeting was focussing on the software threads because they were the ones undergoing the most significant updates in the current review, as the ops and networking papers had been more recently reworked and were currently stable.
   2. TR noted that with the increasing momentum of the devops model, students aiming to be sys admins or DBAs would still benefit from the proposed new model for the software engineering paper.
   3. RT felt that the title "Software Engineering" might be intimidating, especially to women, and that a change would be constructive. She noted that many of the indicative content elements were not specific only to software developers, but would be relevant to managers or business developers.
   4. Discussion of possible new course title; proposals include "Industry Best Practices" and "Industry Fundamentals".