Using Return on Investment to Compare Agile and Plan-Driven Practices in Undergraduate Group Projects

P.J. Rundle
Department of Computer Science, Heriot-Watt University
Edinburgh, EH14 4AS, UK
+44(0)131 449 5111
paul@paulrundle.com

R.G. Dewar
Department of Computer Science, Heriot-Watt University
Edinburgh, EH14 4AS, UK
+44(0)131 449 5111
rick@macs.hw.ac.uk

ABSTRACT
In this paper we describe our experiences of introducing agile practices into undergraduate group work by comparing the results to more traditional plan-driven groups. When considering whether to adopt an agile or plan-driven project management strategy in a commercial context, Return On Investment (ROI) is an important factor. We have adapted the ROI model to our analysis to assess what affect a chosen development approach has on the outcome of the groups’ projects. In our investigation we observed seven software teams as they implemented a business information system. Two groups adopted agile practices, including fortnightly iterative delivery; the other groups were controls. We found that being labelled agile did not necessarily imply that a group’s practices were more agile. Also, it was unclear whether the so-called agile groups delivered a better ROI than their plan-driven counterparts.

Categories and Subject Descriptors
K.3 [Computing Milieux]:COMPUTERS AND EDUCATION
K.6.1 [Computing Milieux]: MANAGEMENT OF COMPUTING AND INFORMATION SYSTEMS - Project and People Management

General Terms
Management, Measurement, Economics, Experimentation.

Keywords
Agile, plan-driven, undergraduate groups.

1. INTRODUCTION
Many project management techniques and methodologies have fallen in and out of favour in the field of information systems development during the few decades in which IT has been applied to business problems [1], [2]. Early development projects (pre-1960s) were typically technology-centric and though the programmers employed were well suited to solve the low-level problems of the day, it became necessary to conceive formal methodologies incorporating best practice as the scale and risk of software projects grew. Structured methods emerged in the 1960s and 1970s, leading to the acceptance of the Systems Development Lifecycle [3]. Prescriptive methods such as SSADM appeared in the 1980s, driven by failing projects in the public and private sector. However, it became clear that these techniques were not always effective as two issues arose. First, a heavyweight approach expended significant effort in areas which do not contribute to the final product, for example comprehensive technical documentation. Secondly, an inflexible engineering approach does not recognise that often the only thing that never changes in a competitive marketplace is change itself [4].

In ‘Balancing Agility and Discipline’, Boehm and Turner divide the methodologies prevalent today into ‘agile’ and ‘plan-driven’ [1]. Since these two terms seem to be the least misleading we have come across, and the most suited to this investigation, they shall be used throughout. Agile methods include XP and DSDM, whereas plan-driven methods such as SSADM are typified by their waterfall lifecycle.

The Department of Computer Science at Heriot-Watt University has, for many years, run a group project in the penultimate (third) year of the undergraduate course from October to May. Students are pseudo-randomly placed in one of (in 2004/2005) seventeen teams. Consideration is made of each student’s previous grades in order for the groups to be of similar mixed abilities. Every student has already completed courses in various disciplines; including Java, databases, HCI, logic, networking, business, etc. Groups consist of 6 or 7 students and each group is given a different fictitious system specification, for example, a booking system for a garage or a production control system for a manufacturing company. Each group is allocated a proxy customer from a member of academic (faculty) staff. The customer clarifies requirements and is involved in the assessment. Each group is also allocated a nominal “project manager” from academic staff that may offer further guidance and also contributes to the assessment. However, this project management role is generally peripheral and we make no further reference to it in our study. Assessment of, and feedback on, this work is triggered by a first deliverable in December (primarily requirements documents).
requirements planning, risk assessment, prototyping and software incremental approach to development, involving cyclic stages of had published the Spiral Model in 1988, which proposed an Amid mounting resistance to the plan-driven approach, Boehm delivered artefacts [5].

in the implementation stage and to improve quality in the need for effort in the planning activities to minimise setbacks with extensive documentation. Supporters of this approach assert the notion of ROI by investigating the quantifiable impact of the groups’ developments on their customers’ expectations and off-setting this value against estimates of effort.

We continue by describing the background to the competing methodological approaches and some related studies, before we mention the procedural and ethical considerations that underpin our own study. Finally, we present our findings and suggest future work.

2. BACKGROUND

Traditional, plan-driven methodologies are allied with engineering, advocating thorough requirements gathering, system design and implementation in rigid sequence; all accompanied with extensive documentation. Supporters of this approach assert the need for effort in the planning activities to minimise setbacks in the implementation stage and to improve quality in the delivered artefacts [5].

Amid mounting resistance to the plan-driven approach, Boehm had published the Spiral Model in 1988, which proposed an incremental approach to development, involving cyclic stages of requirements planning, risk assessment, prototyping and software design/implementation [3].

Later, we saw the rise of DSDM, Extreme Programming, etc. as new members of the agile stable. However, these had their critics and were variously described as naïve or ‘quick and dirty’ [2]. To counter such allegations, advocates of lightweight development met and agreed to use the term ‘agile’ to refer to their common values. Hence, in February 2001 the Agile Alliance was born [6].

There is a wealth of literature extolling the virtues and failings of agile and plan-driven approaches. Here are some resources for the interested reader [7], [8], [9], [10], [11]. But our focus here is not to feed this debate; instead we wish to consider what the affect of these two approaches might be on the development work of undergraduates working together in small groups.

Several studies of agile methodologies have been conducted in an undergraduate setting over the last few years. Reichlmayr [12] states that while it is important to understand the waterfall model, such an approach to undergraduate projects denies students the opportunity to experience common occurrences in the software development lifecycle. An agile approach can incorporate events such as requirements change and allow students to appreciate process improvement and refactoring.

Reichlmayr’s study was conducted at the Rochester Institute for Technology (RIT) using undergraduates and has a lot in common with our own study. RIT’s project is compulsory for second year students. Teams of five or six are allowed, but in contrast to our students, they can choose their team mates. Although many students go on to study software engineering in greater detail, for some this is their only experience of a development project within RIT. Again, this is in common with our students.

As is the case in Heriot-Watt, the RIT project conventionally follows a waterfall approach, although the duration was only one semester; considerably less than the Heriot-Watt project, which lasts over 30 weeks including Christmas and Easter breaks. In addition, students at RIT did not have the opportunity to gain detailed mid-project feedback. In contrast, our students do get some broad feedback on the requirements and design deliverables.

In designing the project to be more agile, Reichlmayr took a subset of principles from the Agile Manifesto rather than adopting a specific methodology. These included: deliver working software frequently; face-to-face communication; simplicity; and team reflection.

These principles appear to have been selected on the basis that documentation could not be omitted, presumably because it is important for assessment. As at Heriot-Watt, time (delivery date) and cost (number in team) are fixed and it is only the scope that is open to negotiation between the customer and development team. In Reichlmayr’s study, students were required to build unit and acceptance tests and to adopt Eclipse and CVS. In our study, no tool sets are mandated, although coincidentally, instruction is provided on Eclipse and CVS. At RIT, releases were made in weeks five, eight and ten, with week ten being the final deliverable.

A key problem identified by Reichlmayr was the reluctance of some groups to embrace agile behaviour and so work only on tasks required to implement the current iteration. Groups were seen to code ‘hooks’ ready for future iterations, and even include functionality which was not requested by the customer. Unit testing was also difficult to sell, however JUnit was used by some students.

Reichlmayr concludes that agile techniques were effective in the teaching of this course. Delivering iteratively provided an opportunity for reflection and continuous integration aided collaboration. Continuous integration raised the students’ appreciation of the need to test throughout and user stories increased awareness of risk management and customer priorities [12].

Germain and Robillard conducted similar research at Montreal University, although this study involved measurement of cognitive activity throughout the lifecycle in order to determine the distribution of effort between activities on RUP-like and XP projects [5]. Students were given a requirements specification which had to be implemented within 45 days. Following this deliverable, a further specification was presented which involved architectural changes to the original system and had to be delivered within 15 days. This research showed that the XP teams spend significantly longer involved with implementation activities

the following February, the second deliverable is mainly design documentation. Final documentation, a presentation and a prototype demonstration are delivered to the customer towards the end of the academic year. Some peer review also contributes to an individual student’s mark. As a result of this assessment regime, the general pattern of development has been, anecdotally at least, plan-driven.

This paper reports on an experimental intervention that enabled us to compare plan-driven and agile approaches in developing business systems in the context of these undergraduate group projects. In both the public and private sectors, Return On Investment (ROI) is an objective measure of success, yet it difficult to find a comparable instrument with which to assess undergraduate development where there is necessarily diversity in specifications, customers, assessors, group dynamics, etc. However, we have adapted the notion of ROI by investigating the quantifiable impact of the groups’ developments on their customers’ expectations and off-setting this value against estimates of effort.

We have adapted the notion of ROI by investigating the quantifiable impact of the groups’ developments on their customers’ expectations and off-setting this value against estimates of effort.

These principles appear to have been selected on the basis that documentation could not be omitted, presumably because it is important for assessment. As at Heriot-Watt, time (delivery date) and cost (number in team) are fixed and it is only the scope that is open to negotiation between the customer and development team. In Reichlmayr’s study, students were required to build unit and acceptance tests and to adopt Eclipse and CVS. In our study, no tool sets are mandated, although coincidentally, instruction is provided on Eclipse and CVS. At RIT, releases were made in weeks five, eight and ten, with week ten being the final deliverable.

A key problem identified by Reichlmayr was the reluctance of some groups to embrace agile behaviour and so work only on tasks required to implement the current iteration. Groups were seen to code ‘hooks’ ready for future iterations, and even include functionality which was not requested by the customer. Unit testing was also difficult to sell, however JUnit was used by some students.

Reichlmayr concludes that agile techniques were effective in the teaching of this course. Delivering iteratively provided an opportunity for reflection and continuous integration aided collaboration. Continuous integration raised the students’ appreciation of the need to test throughout and user stories increased awareness of risk management and customer priorities [12].

Germain and Robillard conducted similar research at Montreal University, although this study involved measurement of cognitive activity throughout the lifecycle in order to determine the distribution of effort between activities on RUP-like and XP projects [5]. Students were given a requirements specification which had to be implemented within 45 days. Following this deliverable, a further specification was presented which involved architectural changes to the original system and had to be delivered within 15 days. This research showed that the XP teams spend significantly longer involved with implementation activities
such as coding, testing and integrating, whereas RUP teams spend more time on pre-coding activities. Whilst this is consistent with general expectations, Germain and Robillard felt that the disparity was smaller than they had anticipated. Furthermore, it was concluded that RUP activities do not reduce the effort spent on implementation. One important factor was the transfer of knowledge from agile teams – who started implementing earlier – to plan-driven teams, specifically in the area of servlet technology; a social issue which may have advantaged plan-driven teams.

In terms of the relevance of introducing agile practices to student group work, Melnik and Maurer believe that it can promote professional skills that might otherwise not be addressed [13].

3. METHOD

We now return to the detail of our own study. After the students had been allocated to groups, all teams were offered the opportunity to be part of this research. In the end, we recruited two groups to deliver in an agile manner and five controls working in a plan-driven manner. This plan-driven behaviour was typical of that which happened in previous years and constituted minimal intervention on our part.

To compare the two approaches, we measured ROI throughout the project lifecycle by evaluating demonstrable functionality alongside customer requirements, versus team effort.

During the period when all groups were preparing their second deliverables, and before development began in earnest, we met the two agile groups to provide more information on how to adopt agile methods and how this would work in the context of the their projects.

Also at this time, we arranged to meet the customers of the seven participating groups to assign values to all aspects of the system. This was achieved by taking the functional specification – produced as part of the first deliverable – and annotating this with values based on the customer’s perception of how much a given feature would be worth to their fictional business. These values were not monetary amounts, as this would have been very difficult for customers to quantify accurately or consistently. Instead, we suggested that customers start by allocating 100 points to one feature, and then assign subsequent features values relative to this. These figures were untimely normalised across all groups.

Agile groups were given a copy of the value-annotated specification to increase their comprehension of the customer’s needs and to be used as a base for planning development. Non-agile control groups were not made aware of these values in order to maintain their plan-driven philosophy of attempting to implement the full system to the specification, rather than prioritising development in an agile manner.

Groups who volunteered to “go agile” delivered iterations in two week timeboxes, whereas plan-driven groups worked to the deliverable schedule with no extra mid-project iterations. Agile teams met their customer at the end of an iteration to assess achievements, receive feedback and prioritise features for the next iteration. Measurements were taken at these iteration meetings to gather results on the ROI realised. An iteration review involved the customer assessing parts of the system that had been developed in the previous iteration and deciding on percentage values for the completeness and fitness-for-purpose of the delivered functionality. A percentage-based approach was easily understood by the customer and group, and gave quantified feedback to the group. It was made clear to the customer that when measuring delivered functionality, the customer should focus on business value achieved rather than level of effort expended by the group. As an anecdotal example, a customer gave 20 percent complete for a sign-up page on one application because, although the functionality was perfect, she was not happy with the design and stated that it would demean the brand.

With respect to cost, a copy of the source code was taken, allowing us to measure level of effort expended during an iteration. Statistics were run on this source code using the SLOC Count tool [14]. This was then used in a COCOMO II calculation to derive a cost figure based on a notional monthly developer cost of £1,708 [15].

In contrast to the two week data collection cycle for the agile groups, a less frequent (6-8 week) cycle was adopted for the five plan-driven control groups. Meetings and measures of effort and value were conducted in the same way as before. However, plan-driven groups were given minimum notice of the assessment point in order that they did not attempt to deliver a working iteration or engage in customer contact that they otherwise would not have had. As it happened, working alpha prototypes were available from the non-agile groups at these stages so that there was a code base to measure costs against and an application for the customer to assess for value. If this had not been the case, a stereotypical zero value would have been recorded by the plan-driven groups at the interim measurement point.

Whilst our aim was for agile and plan-driven groups to work strictly in accordance with their adopted project management style, clearly this variable was out of our control due to individual behaviour and group dynamics. For instance, plan-driven control groups may have naturally, or inadvertently, chosen to adopt agile practices, or their customers may have demanded software delivery in an iterative manner, without formally or consciously subscribing to agile. In acknowledgement of this issue, we gathered data on the level of a group’s agility in questionnaires administered during the measurement meetings. In other words, we have measured agility explicitly, rather than making the assumption that groups who opted for iterative delivery would be the most agile. We asked each group the following questions and corroborated some of the answers with similar questions to the customer to check consistency.

1. As a percentage of the total, how much time did you spend working on the project, face-to-face, with two or more other team members?
2. Did you ever seek to integrate ALL code to view the complete system at a given point?
3. How often did you meet the customer face-to-face?
4. If the customer requested amendments, did you take these on board?

Table 1 shows the agility ratings (0.85-1.15) that were associated with the standard answers the groups provided. Given the lack of a published model suitable for this situation, we have followed the conventions used by COCOMO to inform our choices for these ratings.
Table 1. Level of Agility Rating for Questionnaire

<table>
<thead>
<tr>
<th>Rating Factor</th>
<th>Low agility</th>
<th>&gt;</th>
<th>&gt;</th>
<th>&gt;</th>
<th>&gt;</th>
<th>&gt;</th>
<th>High agility</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q1</td>
<td>&lt; 30%</td>
<td>30%+</td>
<td>40%+</td>
<td>50%+</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Q2</td>
<td>Little integration until end</td>
<td>Some parts occasionally</td>
<td>Most parts regularly, but not as a whole</td>
<td>Whole system often</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Q3</td>
<td>Once a term</td>
<td>Monthly</td>
<td>Fortnightly</td>
<td>Weekly</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Q4</td>
<td>No, stuck to spec.</td>
<td>Took on some requests</td>
<td>Took on many requests</td>
<td>Compromised</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Each group completed one questionnaire at each audit point while they were being assessed for costs and value delivered. The average rating factor for each question was taken and the product of these average answers provided an agility rating for the group over the course of the project.

3.1 Ethical Considerations

The group project contributes to each student’s grade so any discrepancies in marks created as a result of changing the group’s project management strategy has an ethical dimension. However, the project contributes to less than five percent of the total degree grade, minimising any potential downside.

Openly forcing agile methods upon groups would also have opened up the possibility of appeal if the team felt their final grade had been negatively impacted, or competing groups had been positively advantaged. As a result, agile groups were volunteers and were allowed to opt out at any time.

Another difficulty was the participation of non-agile control groups; what would motivate them to be a part of this research given that there was no perceived benefit from doing so? With the agreement of the academic responsible for administering the group projects, the control groups were compulsorily recruited rather than being volunteers. Any disadvantages were considered negligible enough for this approach to be defensible in light of the potential worth of the study.

Finally, intellectual ability of participants is a potentially important factor in team success. For this reason, it was important to measure the intellectual ability of each group. These ability metrics were group averages of each member’s recent grades. Since the group identities we refer to in this paper are not the same as those used by the staff and students at the university, there is no way to match grades to groups, never-mind individuals.

4. RESULTS

As we have said, we understood that the level of group agility was outside of our control. The results from the questionnaires we administered to address this issue are given in Figure 1.

Figure 1. Group agility derived from questionnaires

From Figure 1, the so-called agile groups (B and G) were somewhat “agile”, but the practices and attitude adopted by groups A, E and F exhibit higher levels of agility based on their subjective responses.

Turning now to the value delivered throughout the project, Figure 2 shows that group E outperformed the others, managing to implement almost all customer demands. Groups F and G performed well, reaching around 70 percent of the allocated value. Group A can be seen to accrue value very rapidly between the two measurement weeks in a manner consistent with stereotypical plan-driven development, where the system becomes usable and valuable as components are integrated. Ironically, Figure 1 showed group A as the most agile. So-called agile groups B and G can be seen to deliver ‘quick-wins’ in the early stages of the project, before stalling during March and April, perhaps due to the Easter break. Nevertheless, the value created by these agile groups during January and February supports the view that even early iterations do have some customer value.

Figure 2. Cumulative value delivery throughout the project lifecycle, as a percentage of the total value allocated by the customer

Before we address ROI, the counterpoint to value is cost, and so Figure 3 shows how this parameter varied over time. Here we can see the so-called agile groups are fairly typical in this population and that two non-agile outliers (groups D and E) have incurred higher costs than the others.
To analyse ROI in the context of these student projects, we have made the assumption that the total customer value of each project is the same. Therefore, ROI is the percentage of value realised, relative to the cost of construction. By dividing the percentage value ultimately delivered to the customer by the cost of that development, we arrive at a measure of ROI; namely the amount of value delivered per pound spent. Naturally, a higher ROI is advantageous, as it indicates that the group are able to deliver valuable functionality at low cost.

To illustrate this point, Figure 4 shows the average ROI, with agile group G outperforming their peers. However agile group B did not fare so well with one of the lowest ROI figures. A graph of ROI over the lifecycle was also considered but it gives us little more insight. Visually contrasting Figure 1 and Figure 4 does not show any obvious dependency, although plotting ROI and subjective measures of agility does show a weak positive correlation, even if we discount group G as an unrepresentative outlier.

Due to space restrictions, we can only summarise some of our other findings. For instance, there is a somewhat predictable (although slight) positive correlation between the grade a group received for the project and the value they delivered. There is also an equally slight (although interesting) positive correlation between a group’s grade and their level of agility.

5. CONCLUSIONS

In comparing groups working on undergraduate development projects, we attempted to measure the effect of different development strategies on the outcomes. Specifically, we compared agile and plan-driven approaches. We intervened in the approach of the volunteer groups we designated as agile, but left the control groups fallow, assuming they would adopt a more plan-driven approach.

The first surprising results came from questionnaires where it appeared that being designated agile, and as such having extra tuition on agility, did not make a group exhibit more agile practices.

We then looked at Return On Investment (ROI) as a measure of a group’s performance. This was achieved by comparing the value the project delivered to a proxy customer, against the cost of that value as measured by COCOMO metrics. Whilst independent consideration of value and costs did not highlight any startling differences between the two types of group, when we looked at ROI, one of the two so-called agile groups did seem to deliver a much higher ROI than the other groups. Of course, this value may have been a chance and unrepresentative outlier, particularly considering the other so-called agile group had one of the lowest levels of ROI in the sample.

One explanation of the apparent similarity of outcomes for the two approaches could be that some inter-group communication happened that we were not aware of. Also, the control groups would have been aware that something was afoot during the recruitment phase of the study. In addition, all students had been exposed to some classes on agile methods during their time with us leading up to this project. Any one of these factors could have biased the results.

However, as observed by Hirsch [16], “Agile development is best viewed as an evolution of plan driven development and not as a radical departure from previous practices”. In other words, we should not be so surprised there is less of an impact on the work of the students than we had initially assumed.

In terms of future work, it would certainly be advisable to repeat this study with a larger sample size. Since we have some evidence that there is no major impact on the outcomes, next time it might be more ethically acceptable to instruct a larger number of groups to adopt an agile approach, rather than relying on volunteers. Also, we did not get a summative appraisal of the students’ experiences, as per Melnik and Maurer [13]. This would be useful in future studies to capture a qualitative dimension. Some measure of group coherence or team spirit might also help explain our findings.

From an educational perspective, we hope that the study has raised awareness amongst students and that it might indicate implications to educators. Although we feel it is too early to propose ROI as an assessment instrument, we found no evidence to suggest it would not be a fair measure if used appropriately.
6. ACKNOWLEDGMENTS
Our thanks go to all the student participants for their cooperation. Thanks also to Helen Ashton, Patrick Green, Andrew Ireland and Nick Taylor for their support and advice.

7. REFERENCES