The Agile Semester: How we used Agile as an engagement mechanism

Leigh Griffin, Brendan O'Farrell, Colm Dunphy, Pete Windle, Eamonn de Leastar

Abstract Red Hat, a global leader in Open Source technology, have collaborated with Waterford Institute of Technology (WIT), who launched a world leading ICT conversion higher diploma exclusively online, to help share our industry best practices. The expertise in leading Agile transformations in a distributed environment has seen an accumulation of knowledge around both the best practices that software engineering teams should follow and the pitfalls to remote engagement to lead a process and cultural change. In Red Hat, we experienced this at both a people and process centric level to ensure that lasting change is embedded within the people and teams. Partnering with our colleagues in WIT, the shared knowledge has shaped the evolution of the delivery of the diploma over the last 4 years. In this chapter, we explore how remote Agile can be implemented and the failures that can inhibit a long lasting change. We explore the relationship between key lean tools and the evolutionary impact it has had on our adoption of the Scrum framework, the most popular process improvement approach in industry. Finally, we view this journey through the lens of several years of delivery of an online course which we use as our case study

Key words: Academic Agile, Engagement, Industry Influence, Scrum

1 Introduction

In 2001, the Agile Manifesto [1] was published to establish a set of guidelines for software development best practices. In the 20 years that have passed since its publication, frameworks, tools and methodologies have emerged to fall under the umbrella

Leigh Griffin, Brendan O'Farrell
Red Hat, e-mail: {lgriffin,bofarrell}@redhat.com

Colm Dunphy, Peter Windle, Eamonn de Leastar
Waterford Institute of Technology e-mail: {cdunphy,pwindle,edeleastar}@wit.ie

of Agile. Scrum, one of the most popular frameworks that embodies the mindset of Agile development has become ubiquitous in the software industry. The ability to quickly iterate on ideas, delivering the most value with a customer centric mindset has transformed the customer experience. A side benefit of this is the movement of Software Engineering into a bimodal world, with the traditional approaches no longer suiting the expectations set forth by customers, expectations that have been enabled by the pervasiveness of computing and the desire for innovation. Red Hat, a global leader in software development has long embraced an Agile way of working. The value that Red Hat teams and by extension our customers, have encountered is at the software tool chain, where a resilient, high quality offering has emerged as a systematic approach to creating software the customer centric way. The benefits attained from the production of software products in this manner is no different than any other company experiences. That approach is often termed mechanically Agile, showing that teams are capable of following a process that can predictably deliver a working output, in rapid iterations and with the customers needs in mind. However, the strength of an Agile way of working really only emerges when it is embraced holistically and becomes a process that is both philosophical and cultural, as it becomes a mental mindshift for those embracing it. Beyond that, the Human Resource Management (HRM) perspective, which has received increased academic study, has improved retention, accelerated competency development and created an engagement culture within the teams that practice it. Red Hat has a storied history of community involvement and in Waterford, the Red Hat office has close ties to the local college, Waterford Institute of Technology (WIT). Those ties include guest lectures, participation in industry boards to help shape future modules that are industry relevant, through to the establishment of a hiring pipeline where interns and graduates can find employment. Within WIT, numerous computing courses are available and in recent years, a course designed to transition skilled graduates in other disciplines into software engineering roles was brought online. That course, the Higher Diploma in Software Engineering (HDip), is designed to address the skills shortage in the IT industry and is heavily influenced by industry in the local region. A defining influence for the HDip was the embracing of an Agile mindset at both a practical student focused level and a philosophical level by the lecturing staff. This paper explores the industry perspective on Agile that influence the adoption of Agile approaches in the HDip and charts the experience from both a student and lecturer perspective as this new paradigm takes root.

2 Agile: An Engagement Driver

According to [2], one of the biggest challenges in academic literature, when it comes to engagement, is the lack of a universal definition. Engagement has been defined as the emotional and intellectual commitment to the organisation and as noted by [3], connects employees with cognitive and behavioral consequences that help the organisation's goals. [4] defines engagement as the umbrella term that includes a myriad

of organisational efforts to involve stakeholders in organisational activities and decisions. When defined this way engagement is often interchanged with involvement and dialog. [5] notes that employee engagement drives a number of positive organisational outcomes such as profitability and [6] views it as a competitive advantage. [7] observed key drivers of engagement which included meaningful and challenging work, autonomy in decision making, career opportunities, organisational concern for employee wellbeing, and sense of feeling valued and involved. Engagement is a continuous process and [8] stressed the need to maintain an engaged workforce by implementing strategies to build a positive and supportive work environment. This includes facilitating employees with appropriate resources and crucially, continuous monitoring to keep out ambiguity in the workplace. The author notes that this can produce a productive workforce which in turn leads to sustainable organisational success. Formal models have been put forward to quantify Engagement. [9] defined the Kahn Needs Approach and [10] the job demands-resources model. All of the models share characteristics and similar drivers that can be distilled down to a willingness to engage in discretionary effort. This discretionary effort is something which [11] calls organisational citizenship behaviors, a trait which is highly desirable. The authors in [12] state that communication is a part of the organisational context in which employees are engaged or disengaged. Holistically considering the definitions, the proposed models and the relevant impacts it can have on an organisation, it is clear that engagement is both industry and process neutral. That is to say that Agile and engagement is not a specific paradigm in and of itself, however, [13] notes that engagement is a key factor in an organisations performance and citing [14], they state that engagement is crucial in the successful implementation of any new dynamic changes. Agile is the embodiment of a dynamic change in an organisation, therefore, there exists a nexus point where a focus on the drivers of engagement can become a central theme of the Agile implementation. Engagement is often driven by and enabled by the role of the manager.

Within [15], the leader in people-oriented management controls the behaviors of the team members by referring to their need for virtue; he or she is not an arbitrary administrator of tangible benefits, but endeavours not to allow the *rat race* to commence by stressing that there are no better and worse employees and that each worker is endowed with a specific talent. On the contrary, the leader in non-people-oriented management primarily administers the benefits and decides who deserves them. The need for managing people at work and the formal human resource function can be traced back to the Industrial Revolution when the factory system developed ([16]). Working conditions had an imbalance, with respect to pay and hours worked, and during this period of time, according to [17], worker safety was an issue. They note that management were expressing little concern for the safety of their workers and citing [18], whom noted that force and fear were a controlling mechanism. [19] noted that around this time, personnel management, the progenitor of Human Resource Management (HRM), emerged and according to [20] this was attributed to the formalisation, standardisation and adoption of policies and practices to manage

people in work organisations, often driven by legislation. A driving force behind such policies was worker wellbeing through improving health and safety as [21] notes that a social element exists whereby the improvements are not just the resolving of a technical issue, but a key component in the power relationship between manager and employee. The work of [22] on Scientific Management created a world of specialisms by breaking down jobs into discrete parts and training people to become an expert on one component while being part of a larger manufacturing initiative. It can be argued that software engineering is no different, with specialist roles required to complement each other to ultimately deliver a working software product [23]. An often overlooked part of the Talorist philosophy is the boredom of repeating the same role ad nauseam, with work by [24] observing that individuals experiencing this are less likely to engage with or focus on their work and job satisfaction can suffer. The challenge therefore is to engage people through a combination of HRM practices and an underlying methodology that allows the specialism to contribute positively and in a growth manner. Agile, in software engineering, has filled that gap to bridge the Talorist problems that knowledge workers can experience. This happens through whole team delivery, with a multitude of roles and unique skills needed to achieve the teams goals. The ability for the team to self organise allows team members to wear multiple hats on a given day. This sees them take on a variety of work as well as time to observe, mentor, coach, encourage their peers and effectively self direct their work towards a common team oriented goal. This self organisation is a key hallmark of Agile teams.

3 Incremental Delivery of Value

One of the more attractive aspects of delivering software through the usage of Agile techniques is the rapid iteration and delivery of incremental value to the customer. The idea here is to deliver the most value, in the shortest timeframe possible, to allow the customer a tangible, potentially workable feature. This approach allows for flexibility in the overall delivery plan, as the learning generated from the creation of the outputted artifact, as well as the key insights gained from the customers usage, allows for a rapid feedback loop to occur. That results in changes to the Backlog, that is the prioritised queue of work that represents all of the features and requirements to deliver an overall working product for the customer. This is in contrast to the waterfall model of working. While the waterfall model has it's place within the software industry, it has a heavy amount of up front analysis that leads to a technical plan that will be implemented on a set schedule, with minimal customer involvement. That way of working creates domain knowledge silos and deep specialism to achieve each stages goal. That is an antipattern for Engagement and crucially how users consume technology has led to the need to rethink the approach, with the Agile movement becoming dominant in the past decade. A benefit for the team tasked with delivering the project, it is an opportunity to examine the full value stream of delivery, from requirements elicitation through to customer debriefing. This involvement and

ability to alter the course of the delivery plan facilitates an engaged environment where their input is both valued and desired. However, Agile, as defined by [25], as a way of managing projects, where that work is divided into a series of short tasks, with regular breaks to review and adapt plans, is not naturally applied to academic courses. Courses are analogous to waterfall projects, typically run in 1-4 year durations and in recent years, the word semester which comes from sēmestris in Latin, meaning *half-yearly* has emerged as a means to increment students towards an overall goal. In modern usage, semester means *one of the two periods that the school or college year is divided into* allowing a more focused delivery in the chosen timeframe. The idea of semesters is a departure from years e.g. first year, second year, etc. where subjects ran for a whole year and is a positive step towards a more Agile centric delivery model. Within these semesters there is further opportunity to move even more closely to an Agile centric delivery model by mirroring the Scrum approach to creating a timeboxed delivery mechanism known as a Sprint. Through splitting the semester into consumable sprints such as 4 x3 week sprints or 3x 4 week sprints, the opportunity emerges to rethink the delivery of value within the confines of a smaller time schedule. This departure from the norm is part of the move to homogenise education and adopt the most popular approach used by universities. It has a passive benefit of de-risking the student experience by providing a smaller and more focused learning experience, with possibilities of interventions earlier in the cycle if the student so needs it. It is also part of the modularisation process, where subjects became modules with defined effort hours and learning outcomes, popularised by the Bologna Process in Europe to enable increased staff and student mobility [26].

4 The Agile Semester

Despite the semester implementation moving the academic delivery towards an Agile way of delivery, it can often lead to a sub optimal experience for both educators and students. Educators are often *flying blind* for as much as the first half of such a semester, unaware as to the level of engagement of their students, relying almost exclusively on the results of an initial assessment to provide emprical evidence about progress. This is typically gathered at the half-way point (the mid term), where traditionally a body of work can culminate in the delivery of an assessment for grading, one that is often not assessed until sometime beyond this point due to workload and student volume. Due to calendar differences, the mid term may not occur after week 6, as the duration of the semester may change due to a combination of start and end dates being fluid and to accommodate National Holidays within the country. At this stage, much of the semester has passed, and any opportunity for course correction, revision or re-expression of core content may have passed. This assessment may also be the students first exposure to the state of their understanding of the core concepts. They are then relying on a rapidly compressed schedule to *cram* for the final assessment – often abandoning more general educational attainment at

this point, driving towards a narrow set of achievements to satisfy the assessment imperatives. All of the above can be even more pronounced in an online only delivery, where the variety of informal cues from in-person classroom delivery are absent.

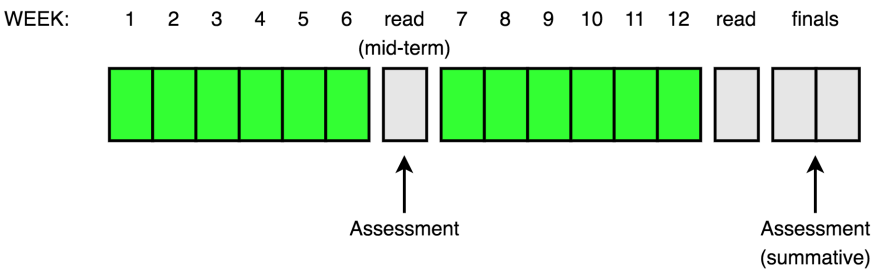

Fig. 1: Conventional Semester

There are significant parallels between conventional waterfall software development methods and the standardised model for educational delivery visible in Figure 1. Both the traditional academic semester and the Waterfall method often exhibit the following characteristics [27]:

- Unidirectional communication
- Pre-ordained schedule
- Single point of failure
- Limited opportunity for course correction
- Highly stressful final deadlines - with consequent impact on quality

The software industry has long since adapted a range of practices to ameliorate the forces that arise in this context and moreover regularly introspect to optimise even further, removing wastes and handling unforeseen circumstances. These have evolved into the Agile set of methods, and often include these practices are:

- Closing the communication feedback loop
- Adaptive schedule
- Early discovery of project status
- Multiple retrospectives
- Growing Software artifacts guided by tests

Additionally, these practices have evolved in the context of distributed or *remote* teams, operating at geographic distance, which puts a greater strain on the communication and coordination to ensure alignment and ultimately success. This evolution have been driven by a cultural shift, as encapsulated in the Agile Manifesto, and also by the emergence of a set of software tools which, when combined with the practices arising from the Manifesto, have delivered significantly more productive

and predictable software projects. The academic community is on a similar journey, gradually realising the importance of a cultural shift to a more adaptive set of practices, supported by a new generation of educational delivery platforms. Such platforms are often termed Learning Experience Platforms (LXP) which focus on putting the student experience first instead of the more traditional Learning Management Platforms (LMS) which are based around learning management [28]. Such new approaches are now required in a world where blended and online exclusive learning is becoming more popular, more accessible and more demanded by students or indeed due to global events such as the Covid-19 pandemic forcing online and blended for public health concerns. Aspects of Agile practices can be deployed into an *Agile Semester* model mirroring key features of the Agile software development pipeline. These include Iteration, Sprint, Retrospective, Release and Figure 2 highlights the concept.

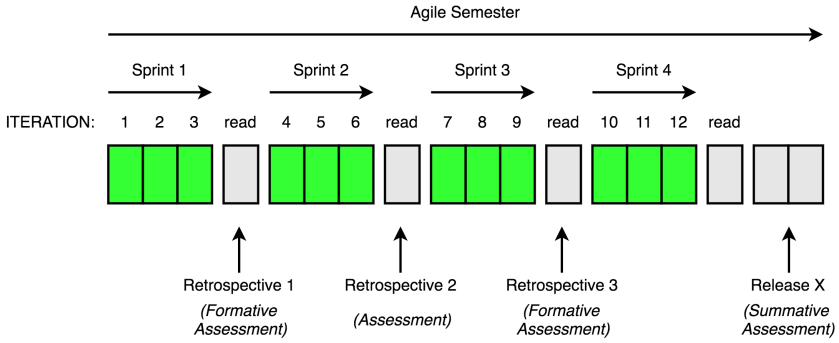

Fig. 2: The Agile Semester

5 Individuals and interactions over processes and tools

One of the core points of the Agile Manifesto is that of a greater emphasis being placed on the role of the individual and their interactions. This often gets translated by Agile teams as an abandonment of processes and tooling that might help support the individual and their interactions. That is despite the wording of the Manifesto being clear, *That is, while there is value in the items on the right, we value the items on the left more*. In industry, the wholesale adoption of issue tracking such as Atlassians JIRA issue tracker, which through smart connectors, can create a development hub, where work can be planned, tracked across the value stream from the Continuous Integration / Continuous Deployment (CI/CD) interactions, to the code repositories through to the teams communication mediums. Tooling is incredibly powerful for enterprises and can complement and support Agile practices, but, if mismanaged, ensures the team ends up following a Tooling Agile approach. Thus, following the

rigidness of the tool instead of the adaptiveness of Agile. In WIT, a balanced approach was taken on the H.Dip in contrast to the generally used Tooling approach.

5.1 Tooling

Central to the adoption of the above model requires a paradigm shift in educational platforms. The software tools which industry have adopted have been critical to the successfully adoption of Agile practices, with many of these tools/practices focusing on enhancing the visibility of all aspects of the project progress to the full development team. This enhanced visibility yields better collective understanding, opportunities for course correction and enhanced opportunities for achieving project goals. This is a core facet of Empirical Process Control, the foundations on which Scrum bases itself, whereby transparency is a key pillar in conjunction with the data oriented inspections that provoke adaptations to the overall process. At WIT, a set of educational delivery components with a focus on learning lessons from the Agile community has been evolved over several years. These components are collectively called TutorStack [30] and are available on an open source license for other institutions to adopt. It is assembled by unbundling key components from various E-Learning platforms, combining them with custom built open source components, and re-assembling the suite into a coherent, loosely coupled platform that supports and empowers the delivery of online courses. Figure 3 shows the overall solution.

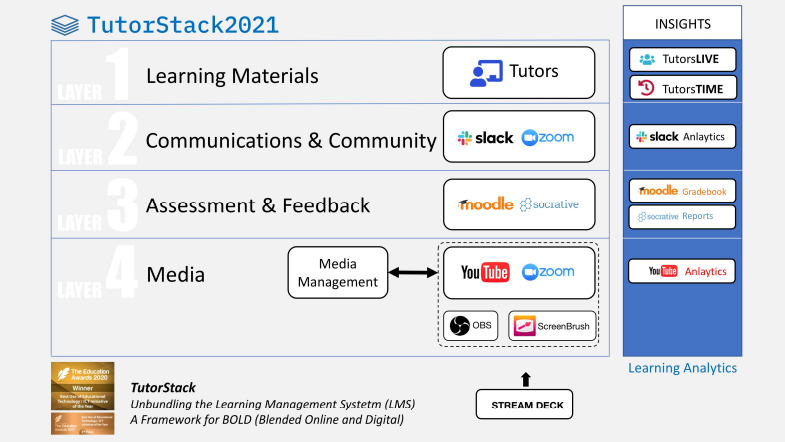

Fig. 3: Tutor Stack

Particular attention has been devoted to enhancing the communications and visibility aspects of course delivery, which is a prerequisite for wider improvements. In particular, ensuring educators and students have clear and simple to digest summaries of student interaction patterns, progress and community behavior TutorsTIME and TutorsLIVE were developed with this in mind.

5.2 Interactions and Engagement

Learning analytics have been a feature of Educational content delivery for some time, with a broad set of standards, techniques and approaches utilised. These range from the highly detailed measurements devised by systems such as Google Analytics, to simpler log files generated by E-Learning platforms such as Moodle. These are at opposite ends of a data spectrum. At one extreme, analytics reports track users interaction patterns in detail, correlate this with data from other sources, and generate a vast range of anonymised page flow, web site performance, and page timing data. At the other end of the spectrum, log based systems track visits to specific pages, or access to a specific resources with simple time/date stamps. Both types of systems can tend to assemble very fragmentary slices of student behavior (highly dependent on content structure), and retain this data in obscure or impenetrable formats. This is rarely made available to either students or educators in a digestible format and where it is made available, the skillset to interrogate and interpret the results may be lacking. As a result, useful insights into student behavior can be challenging to extract.

TutorsTIME is a feature of the Tutors course delivery model to simplify the approach to Learning analytics, constructed around three principles; Simplicity, measurements can be easily explained and rationalised; Transparency, gathered data is shared with the students; Semantically Meaningful, the measurements can quickly deliver useful insights. In Tutors, courses consist of Topics. Topics are made up Units. Units consist of Talks (videos), Slides, Resources, Labs and Archives in the form of compressed files allowing the students a base reference for key stages of the learning outcome. Labs are composed of Steps, Exercises and Solutions. A typical course may have 12 topics, each with several units, which in turn have a mix of talks, slides, labs and resources as above. All of these are presented in conventional web pages.

Measurements Central to the measurements is the course structure maintained by all courses in Tutors. TutorsTIME will simply record the number of minutes each of these pages was active in a browser – not minimised and not in a background tab. This data posted to in real time to the TutorsTIME Data Store. Crucially, the data store encapsulates the hierarchical relationship embodied in the course structure.

TutorsTIME Insight Adapters From these measurements, a set of adapters can be constructed to deliver interesting and useful insights. These adapters are built using the TutorsTIME Data Store and are designed to provide a visual summary of significant student interaction patterns. These insights have a Student Perspective

and an Instructor Perspective, which reveal, respectively, a personal report to each student and an aggregated report of all students to an instructor.

To date, three adapters have been developed: [1]

Lab Progress: Labs are a central feature of typical Tutors courses, and a students progress through the labs is a primary indicator of student engagement. Their lab time is presented as a simple colour coded table, that, at a glance shows their current status in keeping abreast of the course material

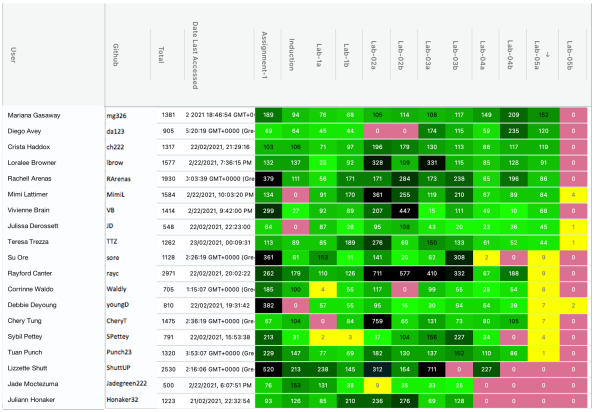

Fig. 4: Heatmap of Lab Progress showing minutes spent and flagging students behind

Academic Calendar Activity: Modelled on the popular approach taken on Gitforges, wherein a profile activity chart for interactions is presented. This shows number of minutes active per week, for each week of the semester, expressed in a simple colour coded heat map.

[1] Figures 4, 5 and 6 show examples of each of the adapaters – with personal details removed. These are Instructor Perspective examples, the Student Perspective presents just individual data to each user. Additional adapters have been prototyped and are under evaluation with respect to their usefulness.

Fig. 5: Academic Calendar Activity, number of minutes per day of week in semester

Semester Profile: A longitudinal perspective on the semester, profiling students online across each semester week, sorted into an *at a glance* relative comparison.

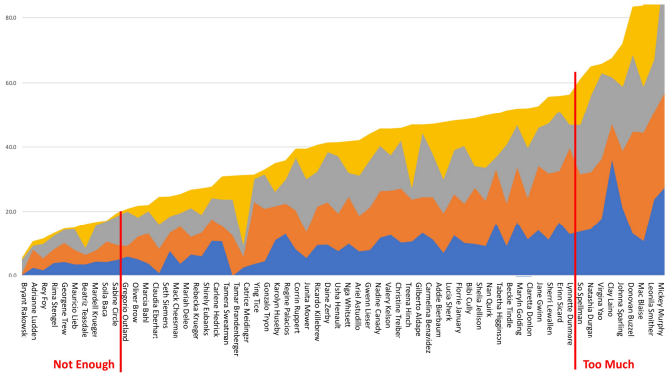

Fig. 6: Cumulative hours online per week per student (4 weeks shown)

5.3 Community Building

Building a sense of community within an online course can be a challenge. In particular, replicating the sense of common purpose associated with in learning activity, particularly in practical subjects like computer programming, often relies on chat / discussion forum applications such as Slack, Discord or Teams. While these are useful, they can be unsatisfactory, as they require students to actively volunteer status and activity, purposefully engaging in chat interactions. TutorsLIVE can partially replicate the serendipity of a traditional *Practical* session, wherein students assemble in a physical laboratory to work on course material, perhaps supervised by an instructor or self-directed. The students can at a glance see who

else is online and what they might be working on, somewhat like looking around a computer laboratory and glancing at the students screens from a distance. They then might choose to initiate a chat/discussion session to reach out if they were interested in collaborating or to seek assistance. Perhaps they may just work away gleaning some community support just by the presence of classmates engaged in a common purpose.

The TutorsTIME StudentCard is the primary visual aid, consisting of 4 quadrants. These are filled with relevant information, which clockwise from top left:

- Student picture or avatar (from github)
- Topic image (from Tutors course topic)
- Lab image (if student is working on a lab)
- Time last active + topic, lab name

Students arriving in a lab (just opening the TutorsLive page on the course web) will see one StudentCard for each student currently online. This will be updated in real time, with the quadrants displaying current course interaction immediately. Students can choose to appear offline via a check box on the top left.

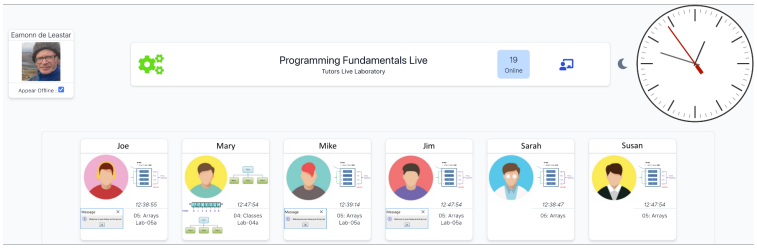

Fig. 7: Tutors Live Lab Session)

6 The Student Experience

From a personal growth perspective, people who work in an Agile team are typically cross functional, self organising and highly motivated to create a work environment that suits their specific skills and abilities. From a HRM perspective, this mode of working facilitates more self guided performance goals and the openness of the team through the mantra of all work being shared and capable of being fulfilled by anyone in the team, removing barriers to participation. The shedding of titles and contractually defined job remits in an Agile team leads to natural empowerment and engagement and within Red Hat, the promotion path is defined not in years spent, but in competencies attained. Agile teams allows for those competencies to mature at a faster rate than their non Agile counterpart, delivering incremental value in the persons career to help them attain a next career step. The Agile philosophy is not just extending to the core skills a modern software engineer needs. The embodiment

of Agile working is effective communication, conflict resolution, self organization and the soft skills required to become leaders. The leader-leader model is how Agile teams work best, giving everybody, irrespective of their experience, an opportunity to lead and thrive. Academically, Agile methods are taught as a means to prepare students for their work placement and ultimately their graduate role. The delivery of the HDip in an Agile manner both at the planning and day to day interaction level, exposes the students to a real implementation of Agile, not a theoretical construct, but rather a tangible experience that they are living through. This appreciation for the process skills coupled with an inherent continuous improvement mindset is leading to stronger graduates working for Red Hat and leading to a continuous feedback loop into the HDip course.

The HDip programme strives to develop the *engaged* student via a combination of the agile semester, the collaborative communication platform and advanced tooling as we have discussed. This multi faceted approach ensures that an environment where engagement can occur is present. One additional aspect, central to any Academic programme, is assessment. Our objective here is to stimulate a key driver of engagement: meaningful and challenging work, combined with a degree of autonomy in decision making as we have outlined above. To achieve this, we marry Agile thinking with the principle of Constructive Alignment [Biggs 2003] in educational assessments. This principle proposes a close alignment between assessment criteria and the learning objectives of a course. Essentially, the students can visualise a clear linkage between the core skills they are acquiring and the assessments rewards they will accrue via the completion of assignments, examinations and projects. When Constructive Alignment is employed in a course, an assessment rubric becomes a critical part of each assignment specification. To be effective, this is often expressed as a two dimensional grid, enumerating individual assignment features, feature sets and grade banding ranges. At a glance, the student can visualise how the course curriculum is mapped to a set of achievable (and graded) goals. Figure 8 below is an example of a rubric for a Mobile Application project from the programme. The column titles identify aspects of the domain as explored in the course content. Each row then enumerates a specific aspect of the domain and associates it with a grade band – from Baseline (pass) to Outstanding (the highest grade).

	Reading	Station	Member	Features	Code
Baseline	Code, Temp Wind Speed Pressure	Station Name Latest weather, Temp C, F, Wind Bft, pressure	None	Load and display stations + their readings from json file + display latest weather for station	Zipped archive +readme
Release 1 (pass)	+ Wind Direction	+ Wind Chill, Wind Compass	None	Dashboard shows station list + button to open station view. Include forms to add new Station + new reading	+ Glitch Project
Release 2 (good)		+Lat, Lng Max/Min (Temp, Wind, Pressure)	First Name, Last Name, Email, Password	Members can signup/log in. Members may create any number of weather stations. Members + sample stations + readings loaded from YAML	+ github repo
Release 3 (excellent)	+ Time/Date	Temp, Wind Pressure Trends	User can edit their personal details.	Member dashboard list summary lists latest conditions for all stations. (alphabetically). Members can delete reports or stations	Github repo with history
Release 4 (outstanding)	Auto Read from OpenWeatherMap API	Station Map View	User can edit their personal details	Simple Graph of Trends	Github repo tags

Fig. 8: Example constructively aligned grading rubric

Students are encouraged to consider a rubric like this in an Agile context: baseline corresponds to a Minimal Viable Product (MVP), with each subsequent band conceived either as an iteration / sprint, or as a release (consisting of multiple iterations), depending on the scale and scope of the assignment. The student then plans their work on the assignment using a classic iteration/sprint/backlog cycle. To complement this, over an assessment sprint, weekly Q&A sessions are held to clarify requirements in the same manner a Product Owner would. A dedicated chat channel facilitates group and class discussion so that all clarifications and communication are both in place and made available to students to examine.

- They can set their overall goal, achieving all aspects at Outstanding level is regarded as a *stretch* goal, not easily achievable for all assignments. So the students must prioritise the assignments/modules where they aim to achieve their highest grades and within each module examine value driven tradeoffs for where they can gain the most experience and the highest marks within that category. This gives the student considerable autonomy in aligning their coursework with their circumstances, such as level of effort to attain a certain score, time available for committing to this aspect of the course and individual skills.
- Each *row* does not have to be completed before advancing to the next. For instance they may have a special interest in Persistence in the above example, pursuing that domain until it is complete, but less interest in UX, so stopping at Excellent in that range. This makes their work meaningful to their personal concerns and learning objectives and creates a custom experience for each student that helps drive their engagement by focusing on amplifying their strengths and interests.

The result of this is the student will feel more engaged in the programme, their choices are meaningful with each student working towards their special interest and it offers the student considerable autonomy with respect to their scheduling and overall targets, all of which are withing their control. To further promote these characteristics, students are encouraged to take the totality of any given semester into consideration, devising a broad workplan to accommodate all assessments in this context. Figure 9 below illustrates this – five assignments E1, E2, D1, D2 and I1 are spread over a 7 month period in this example.

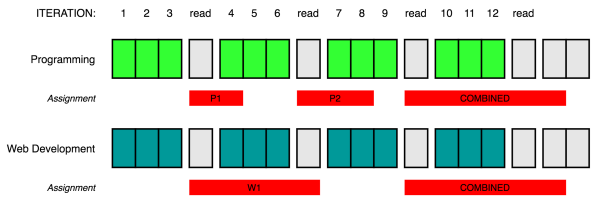

Fig. 9: Example constructively aligned grading rubric

Each of the assignments will have a grading rubric like that shown above in Figure 9. Taken collectively these rubrics constitute the curriculum of the course and can be aligned with the grading achievements the students can expect to achieve, based on their choices (and effort) in each assignment. This structure has parallels with a broader epic/release cycle for an Agile software project. This schedule is a highly visible part of the semester preparation. Additionally, a complete version of the previous year's programme is made available, with the caveat that it may change as the semester proceeds. Thus at the commencement of a semester, the student has some yardstick to assess the effort they may be about to commit to it.

7 Agile Delivery and Planning

The agile methodology is also utilised across two discrete phases during the planning and delivery of this programme. While applications are being processed and candidates are being interviewed, planning for each new intake begins (at least two months before the first day of classes). Key stakeholders are involved in this phase that bring all the required skills to the team in order to succeed. This is analogous to the Scrum Team formation with a mix of skills needed. While Scrum is not prescriptive on titles, it defines exactly three roles, the Scrum Team itself needs to comprise of roles and skills from various parts of the organisation to allow the skills mix . Comparing in Table 1, we present the mix of roles, which each brings specific skills, for both the academic team and a typical cross functional Scrum team:

Academic Team Roles	**Software Engineering Team Roles**
Course Leaders	Technical Lead
Lecturers on the First Semester	Developers, QA/QE
Teaching Assistant	ScrumMaster (Servant Leader)
Online Engagement Officer	DevOps Engineer
Center for Technology and Enhanced Learning	Customer Representative (Product Owner)

Table 1: Analogous Skills between Academic and Software Engineering Scrum Teams

The online engagement officer is a novel role in that it is a dedicated representative for the student experience and for examining student sentiment. The role tries to encapsulate an on campus experience such as having virtual coffee morning as well ice breaking and community building events. The role combines this with elements of coaching individuals to help with mindset and confidence issues as they transition into a new role and career.

Weekly sprints are managed on a digital Kanban board (Trello), with over 120 cards across 19 lists. This allows the capturing of actions around new student registration and admissions, the induction event, programme handbook updates, academic calendar, assignment conflicts, learning management systems, instant messaging workspace organisation, eLearning delivery studios and course material. In addition to the technical skills being learned, transversal skills are a key element of each assignment, as students must write reflections, self grade and present their work in the form of a video and sometimes attend a remote video interview for further questions and clarifications. Task review, allocations and actions are all captured here as well. Knowledge and observations are also captured in Trello. Each year this Trello *Board* is duplicated providing a loose blueprint for the planning phase, with the knowledge and observations from the previous year providing clarity and context on any decisions made. Many issues that could crop up during the academic year are identified in this phase, it is essential for the smooth running of the programme and de-risks the delivery greatly. Having all the key decision makers available in the weekly sprints vastly speeds up the entire process.

Delivery Sprints occur once the programme has commenced. Here, a subset of the programme team meet for weekly sprints on student engagement, retention and knowledge exchange on delivery platforms to improve practice. The core element of these sprints is to use metrics and learning analytics to identify students at risk of dropping out, this is crucial in the first three weeks of the programme. Initially the only identifiable analytics we had available were:

- If the student had not joined the instant messaging (Slack) workspace
- Number of messages posted
- Last login
- Number of days logged in over the past 30 days

These became key indicators of at-risk students, who without early interventions, would never fully engage with the programme. TutorsTIME has added another important layer of metrics to help us measure how active students are and when we need to make interventions. The engagement analytics are crucial for making early, timely interventions to give every student the best chance of success. A driving factor here is the anecdotal evidence that those who fall 3 weeks behind in a semester are most at risk of dropping out.

8 The Agile Principles: An Academic blueprint

The Agile Manifesto defines 12 principles to assist teams in implementing Agile practices [29]. The HDip proved that we can combine these terms to create an Agile semester. While Management, Unions, and Lecturers had mixed feelings and concerns, they could see opportunity once the heart of these concerns, change resistance, were identified. These are the same change management issues experienced throughout the business world for decades and are no different when it comes to change in an educational institute. Most people are averse to change, but to implement Agile practice even in a pilot project requires buy in at all levels. Luckily, this was achieved and the interpretation of the 12 values that were followed within the HDip are now serving as a blueprint for more courses within WIT to come on board with the approach and model. The values, modified slightly to remove the emphasis on software domain, and our interpretation are as follows:

Agile Principle	**Academic Blueprint**
Satisfy the customer through early and continuous delivery	TutorStack established, embracing of technology and delivery mechanisms to put the student experience first
Welcome Changing Requirements	Transition from Adobe Connect to YouTube, enabling mobile viewing, on demand playback, offline offerings, chunky videos
Deliver Frequently	Continuous changes to tech stack in short feedback loops
Work together daily through the project	Lecturers and Students work hand in hand to improve the experience on a daily basis Self policing of new roles and duties
Build Projects around motivated individuals	Project sponsor (Head of Department), enabling support structures freeing up the key drivers
Face to Face Interaction	Scrum Ceremonies followed by the lecturing staff to plan, prioritise and record progress
Primary measure of progress	Retention is the measure of success. An Engagement Officer appointed to the lecturing team
Sustainable, constant pace	Consistent burndown of course material, flattened bottlenecks Business risks identified as course grew
Technical Excellence and good design enhances Agility	Crowdsourcing feedback from students to address deficiencies, world leading software solution invented in house Course Material part of a CI/CD process
Simplicity, maximizing the amount of work not done	Reuse of video recording, async delivery opportunities, rapid delivery of new module content possible through Tutors Consolidation of recorded materials for simplicity
Best designs emerge from self-organising teams	Core team of 3 working on daily improvements
Reflects on how to become more effective	Regular retrospectives and industry touchpoints with Red Hat

Table 2: Agile Principles as an Academic blueprint for wider adoption

9 Conclusion and Future Work

The symbiotic relationship that Red Hat and WIT have established has led to world class graduates, exposed to a delivery mechanism that is inherently Agile and whom fundamentally understand the purpose of it, as an approach to not just building software, but to continuous improvement. The success of the HDip course is now being replicated across more computing courses in WIT and in the spirit of openness, the secret to its success is being shared with other Irish and European institution, with the team regular speakers at education centric conferences. Like any Agile journey, the destination is rarely reached. Each day and each week brings new challenges to continuously improve and reset the horizon goals that were once aimed for. The rest of this section outlines some areas of future work that we wish to explore as we continue this journey towards an Agile Institution.

The four layers of TutorStack encompass Learning Materials, Communications & Community, Assessment Feedback and Media. Through pursuing an *unbundling* approach, selecting a best of breed set of components and services at each layer (with open source solutions preferred if available), has given a springboard for future innovations. Currently each of these operates as an island with separate authentication regimes and user management procedures. Unbundling makes such silo like behavior inevitable, although we believe the merits override any management overhead. However, we can take steps to integrate aspects of user experience across the stack, and our first objective in the near future is to integrate *user presence* across the Instructional Materials (1) and Communications and Community (2) layers. This integration will be via a presence service, whereby status in TutorsLive and instant messaging tools will be synchronised. Thus the students will have an overlapping status across instructional material and the communications platform, further stimulating opportunities for casual interaction and also potentially group work. This is facilitated via simple APIs and via the nature of Tutors platform architecture.

Adding a design sprint to the beginning of the planning phase will give us the opportunity to review the previous year on the programme and map out future delivery mechanisms, curriculum, and retention strategies. Utilising a design thinking iterative process with daily sprints for the following five key areas:

- Explore: in this phase we will map existing business challenges, SWOT, external analysis, industry trends, competitor best practice to understand the ever-changing world of online delivery.
- Reframe: We will interpret the insights from the Explore stage and translate this into meaning for our own context. This helps us identify the key challenges and opportunities for the upcoming academic year.
- Ideate: The team will brainstorm and create numerous solutions to the problems identified in the reframe phase
- Visualise: Here we will converge on solutions that are achievable given time and resource constraints.

- Strategise: We will roadmap the implementation of solutions identified in the Visualise phase over the next 7 weeks of the planning sprint.

Open unlocks the worlds potential is a rallying cry that Red Hat uses to promote the open source way of working. In that spirit, Tutors and the TutorStack framework is made freely available as an Open Source program [30]. This is our gift to the academic community and the hope is to build a true community around it, with updates planned for later this year to build out community communications in order to enable other institutions adopt, adapt and expand the solution with the support of others.

Acknowledgements WIT would like to acknowledge the funding support from both Springboard and the National Forum for the Enhancement of Teaching & Learning for enabling the HDip and continued research on TutorStack. We would also like to express our gratitude to their peers in Red Hat for the time and technical insight into how to bring such best practices into the HDip course

References

1. Beck, K., et al. The Agile Manifesto. Agile Alliance. http://agilemanifesto.org/ (2001)
2. Baumruk, R. The missing link: The role of employee engagement in business success Workspan, 2004, 47, 48-52
3. Jiang, H.; Luo, Y. & Kulemeka, O. Social media engagement as an evaluation barometer: Insights from communication executives Public Relations Review, 2016, 42, 679 - 691
4. Lewis, L. K.; Hamel, S. A. & Richardson, B. K. Communicating Change to Nonprofit Stakeholders: Models and Predictors of Implementers' Approaches Management Communication Quarterly, SAGE Publications Inc, 2001, 15, 5-41
5. Saks, A. & Gruman, J. What Do We Really Know About Employee Engagement? Human Resource Development Quarterly, 2014, 25
6. Kular, S.; Gatenby, M.; Rees, C.; Soane, E. & Bailey, C. Employee Engagement: A Literature Review 2008
7. Robinson, D.; Perryman, S. & Hayday, S. The drivers of employee engagement 2004
8. Bhuvanaiah, T. & Raya, R. Employee Engagement: Key toOrganizational Success SCMS Journal of Indian Management, 2014
9. Kahn, W. Psychological conditions of personal engagement and disengagement at work Academy of Management Journal, 33, 692-724, 1990
10. Bakker, A. & Demerouti, E. The Job Demands-Resources model: state of the art Journal of Managerial Psychology, Emerald Group Publishing Limited, 2007, 22, 309-328
11. Little, B. & Little, P. Employee engagement: Conceptual issues Journal of Organizational Culture, Communications and Conflict, 2006, 10, 111-120
12. Bakker, A. B.; Albrecht, S. L. & Leiter, M. P. Key questions regarding work engagement European Journal of Work and Organizational Psychology, Routledge, 2011, 20, 4-28
13. De Alwis, A. & Weerasuriya, N. Impact of Employee Engagement on Lean Manufacturing : An Empirical Study in Sri Lanka FIIB Business Review (FBR), 2017, 6, 33-42
14. Mann, D. Creating a Lean Culture Productivty Press, 2005
15. Baran, M. & Sypniewska, B. A. The Impact of Management Methods on Employee Engagement Sustainability, 2020, 12, 426
16. Kaufman, B. The Development of HRM in Historical and International Perspective The Oxford Handbook of Human Resource Management, 2009
17. Deadrick, D. & Stone, D. Human resource management: Past, present, and future Human Resource Management Review, 2014, 24, 193-195

18. Slichter, S. The turnover of factory labor Appleton, 1919
19. Fleming, S. From Personnel Management to HRM: Key Issues and Challenges 2000
20. Thornthwaite, L. The origins of personnel management: Reasserting the public sector experience Journal of Management History, 2012, 18, 312-330
21. Jonathan, G. K. & Mbogo, R. W. Maintaining Health and Safety at Workplace: Employee and Employees Role in Ensuring a Safe Working Environment Journal of Education and Practice, 2016, 7, 1-7
22. Taylor, F. W. The Principles of Scientific Management McMaster University Archive for the History of Economic Thought, 1911
23. Gurcan, F. & Köse, C. Analysis of software engineering industry needs and trends: Implications for education International Journal of Engineering Education, 2017, 33, 1361-1368
24. Cleary, M.; Sayers, J.; Lopez, V. & Hungerford, C. Boredom in the Workplace: Reasons, Impact, and Solutions Issues in Mental Health Nursing, Taylor & Francis, 2016, 37, 83-89
25. Griffin, L. Implementing lean principles in scrum to adapt to remote work in a COVID-19 impacted software team. Lean and Agile Software Development, Cham, Switzerland:Springer, vol. 408, 2021.
26. Erik E. Lehmann, Michele Meoli, Stefano Paleari, Sarah A. E. Stockinger. (2020) The role of higher education for the development of entrepreneurial ecosystems. European Journal of Higher Education 10:1, pages 1-9.
27. Van Casteren, W. The Waterfall Model and the Agile Methodologies : A comparison by project characteristics 2017
28. Finch, S. The LMS Vs LXP, which one should you choose? [online] https://thinqi.com/resources/blogs/the-lms-vs-lxp-which-one-should-you-choose
29. Principles behind the Agile Manifesto. [online] Available at: http://agilemanifesto.org/principles.html
30. Tutors Code Base [online]. Available at https://github.com/tutors-sdk/tutors-svelte