Successfully Transitioning a Research Project to a Commercial Spin-out using an Agile Software Process

Phelim Dowling
TSSG Waterford Institute of Technology
Carriganore, Waterford
pdowling@tssg.org
http://www.tssg.org

Abstract

The ultimate success of any research activity is to see it bear fruit in terms of real life use and commercial success. A key element in driving a good concept or idea through the various research and development stages and into full commercial use is the software process that supports it. In the early days of its evolvement the product will require less in terms of unit test coverage and automated test packages and more in the way of room to research and discover the innovation that will make the product unique and of high value. However, as the project progresses and a horizon appears with capital investors and large customer bases, the supporting software process needs to adapt fluidly to these evolving requirements. Efficient use of resources, shorter release cycles and better levels of quality coverage are a necessity to meet stakeholder demands for new features, better features and all to be delivered more quickly. This paper examines the successful transition of a research project to a fully fledged commercial entity with an emphasis on the software process and quality methodologies used.

Keywords


1 Introduction

Today we live in a world grappling with financial instability, recessions and high unemployment. The value of innovation and research as a contributor to moving society forward and beyond the downturn is well understood [1]. However, the funding streams to drive this activity are not as bountiful as they were previously. Any new research activity has to have a well thought out goal of achieving demonstrable benefits in terms of societal or economic improvements. It will struggle to win the necessary funding without this. To this end, in the area of ICT, a proven software process for delivering real life results from a funded research project is essential. Well established practices from the Agile methodologies can be adopted and applied to a project from its early days and then used to drive the concept through to a commercially viable entity and thus delivering the tangible impact required [2].
1.1 Background

The TSSG (Telecommunications Software and Systems Group) [3] is an internationally recognised centre of excellence for ICT research and innovation. The organisations work is undertaken across core Research Units (RU’s) relating to a range of themes such as:

- Experimental Facilities Management
- Bio-Inspired Networking
- Data Mining & Social Computing
- Mobile Platforms, Messaging and Middleware Technologies
- Security
- Financial Information Systems

The activities within these units vary from basic research to applied research to commercialisation projects that deliver innovative and maximum impact products to customers and partners. This paper will focus on a real case example of an idea that was initially funded as a research project within the TSSG and eventually developed to the stage where it attracted significant venture capital funding. Securing this finance created the space for the project team to spin out as a separate commercial entity and create new jobs for the region.

1.2 The project and the company

The project and thereafter the company that journeyed through this process is FeedHenry [4]. The platform put forward by FeedHenry supports the development, deployment, integration and management of secure mobile apps across the entire organisation. Their system allows for the creation of mobile apps using standard web technologies (HTML5, JavaScript, CSS) in a cloud hosted environment followed by deploys to the main mobile platforms (iOS, Android, RIM, WP7, Nokia’s WRT) from the one build.

This success story would not have been feasible without a supporting software process behind it. This process is championed by the Experimental and Facilities Management (EFM) group within the TSSG. The EFM team provide services to the other RU’s within the organisation in terms of verification and validation (V&V), testbed management and design and usability. With dedicated resources from this team embedded in the FeedHenry project from very early on, the evolution from funded research through to spin out company was skilfully managed and serviced.

Having a process that evolved as the product matured enabled the project to meet its goals and maintain expected quality standards from early research days, to aspiring spin-out and onto fully established commercial unit. We will explain in the next number of sections the strategy employed to achieve this significant transformation.

2 The Research Project Days

The TSSG has a well defined and published agile software development process [5] that has been championed across research and early commercialisation projects by their internal EFM team since 2005. It incorporates key aspects of the agile methodologies and promotes a team based approach where all team members are capable of fulfilling the processes and practices. Some of the key agile tools used within the TSSG include:
• Early, active and regular customer engagement
• Short iterations (2 weeks) with engineering releases every 3 months
• Test driven development
• Continuous integration
• Pair programming
• Code reviews
• Aggressive refactoring
• Daily stand-ups
• Bi weekly iteration planning meetings
• Usability sessions
• Strong defect management process

All of these ingredients add significant value to the work produced from the TSSG. The FeedHenry team and product benefitted hugely from this early grounding in software process adherence. This solid foundation made it successful as a research project and set it on the right course for future commercial success.

3 Commercial Spin-out

3.1 The need to become lean

As the FeedHenry team expanded and a future picture emerged that demanded more frequent releases of features, a more lean process was required. No longer would the traditional 3 month release cycle of the research project model support the business need to get features to customers on a regular basis. The pursuit of any successful ICT start-up is to stay ahead of the curve in terms of gauging customer trends and then delivering on those expectations before any potential rivals [6]. This drive for a competitive edge inspired a new software development process that made the most efficient use of the resources available and allowed for peace of mind in terms of delivering new features of high quality while also covering regression of existing components. This new process, of course, carried forward all the agile tools and methods that were relevant previously but maybe applied slightly differently.
3.2 A new release cycle methodology

Figure 1: Overlapping 4 week iterations with bi-weekly production releases

It was decided to allocate 4 weeks to each iteration. The first 2 weeks would be focused on the development of new features. The second 2 weeks would be reserved for testing (V&V) and bug fixing. However, the key to this process was to overlap each iteration by 2 weeks. This had numerous advantages in terms of using the available resources efficiently and delivering quality releases on a frequent and regular basis.

The test team were not waiting for features to be released. They could anticipate and plan for the features being delivered and then spend 2 weeks verifying the planned release on a dedicated test branch. When the process was up and running the test team were guaranteed a new iteration release every 2 weeks followed by daily releases with bug fixes.

The engineering team had no down time either. After delivering a set of features for one iteration, they immediately started work on the next set of deliverables. They of course needed to allocate some time to bug fixing within this next 2 week window - for the parallel test phase happening on branch for the previous feature drop.

A new production release with sound process and rigorous testing foundations was deployed and available to customers and stakeholders every 2 weeks.

The idea of overlapping iterations has had some debate within the agile community. Some credible and proven experience has advised against it [7]. The strategy of staggering multiple development iterations can prove problematic to manage. However, in the case of FeedHenry, we are overlapping the development and test iterations in a manner that facilitates easier release planning, establishes a concrete test approach and makes the most efficient use of resources. A similar process has been used successfully by the Google Chrome team [8].
4 The Process Explained

4.1 Trunk and Branch

- The engineering team created features and improved code coverage on trunk in 2 week cycles. Code freeze was on the Thursday @ 2 PM of the 2nd week.
- Branch IRx was then created on Thursday PM.
- The V&V team deployed this new branch to their test environment and proceeded to perform a full test cycle on this branch – also lasting 2 weeks:
  - Functional testing
  - Verify new features
    - and create associated automated test cases for same
  - Verify bugs
    - (a) fixed on trunk in the previous 2 weeks
    - (b) merged from previous branch
    - (c) new bugs raised and fixed on Branch IRx
  - Load testing if required
    - Apache JMeter [9] was the tool of choice for executing load testing on the FeedHenry project
  - Manual testing
    - generally performed for apps on handheld devices
  - Automated testing
    - regressed all existing features across multiple browsers
- At the same time that the V&V team was performing system test on the new branch, the engineering team started new feature development, code coverage improvements, etc. on trunk for IRy. The fact that the majority of V&V resources were focused on the branch meant that a genuine commitment to maintaining high levels of code coverage on the build process was required.
- A candidate release build for branch IRx was identified by the V&V team on Friday @ 1 PM of the 2nd week of test. The final bug fixes on branch should have been completed by EOB on Wednesday.
- Bug fixes were merged from branch to trunk on an ongoing basis. The final merge was performed on Thursday PM before the new branch was created. Late bug fixes implemented on the branch were merged directly to the new branch.
- The production deploy was executed by the Operations team on Monday AM.
- Note: The V&V team maintained 2 test environments. After a candidate release build was identified for a branch, that release was left untouched on environment A. This environment acted as a mirror for production in terms of debugging issues or deploying hot-fixes. Deploys and testing for the next iteration were executed on environment B. The function of each environment switched for every iteration.
4.2 Planning and Execution

Iteration planning meetings were held on the first Friday of every iteration (i.e. every 2\textsuperscript{nd} Friday). This meeting was used to plan the next 2 weeks of development work on the trunk and discuss the test strategy and any known bug fixing to be carried out on the new branch. Targets to increase unit and acceptance test code coverage on the build would also be discussed and agreed here.

Daily standups were run on Mondays, Wednesdays and Fridays – except Fridays where there was an IR planning meeting.

An overall roadmap which looked at the 3 to 6 months plan was also maintained and discussed at staff level.

Bug review meetings were held once per week. Initially the project maintained separate systems for bug tracking (Bugzilla [10]) and for feature planning (Xplanner [11]). As the process was streamlined the disjoint between maintaining these two systems became evident. A migration to a tool that encompassed bug management, feature tracking and time management was required. The tool chosen was Redmine [12] and this proved very successful.

4.3 Features on branch

The design and reasoning behind this 4 week iteration (but 2 week release) process was to facilitate the frequent delivery of new features that would have 2 weeks of development time followed immediately by 2 weeks of full system test. This ticks most boxes for an early stage start-up company in terms of staying ahead of competitors in the race to market while not dropping the ball on quality.

But another benefit inherent in this method of operation is that it incorporates a strategy to release even quicker to market if the need arises. On occasions during the FeedHenry project, a hot requirement would arise that would yield immense benefit to the company if a swift route to a production release could be accommodated. This was achieved by allowing a specific feature to be developed on the current branch in test. It meant that the feature could be live in less than 2 weeks. The use of this fast track to production was used only when critical as it has obvious short cuts in terms of quality. However, in each instance, a risk analysis was performed and the confidence provided by the code coverage stats, automated regression testing and a well managed process meant this was a valid route to a live deploy for very specific requirements.

4.4 Towards continuous delivery

The success of the fast track deploy scenario described in the previous section made the team analyse the requirements that would allow for daily deploys or continuous delivery. Some key improvements to enable this aggressive release strategy were identified. A greater level of code coverage at build time would be necessary. An automated installation procedure to perform rapid deploys to the test environment and also the production system would need to be developed. In conjunction with this rapid deploy system, a reliable rollback capability would be needed to return the system to a known good state should problems arise. Finally, the automated test suite would be required to execute faster with multiple tests running in parallel.
5 Automated Testing

5.1 A fundamental aspect of the release cycle

A key enabler of the new process adopted was a suite of automated test cases that performed functional and integration testing on the core FeedHenry platform. These tests verified > 90% of the features available, across multiple browsers and on a daily basis. They were designed to present a detailed report via email of all the tests executed, their pass/fail status, screenshots of all relevant pages and various other artifacts produced during the test execution.

The benefits reaped by maintaining this automated test suite were crucial to the success of the project and its evolution to spin-out company [13]. It provided early and clear evidence of a broken feature or bug. An accurate sense of product stability and functionality was gleaned as > 90% of functionality was exercised and regressed on a daily basis. It also acted as a safety net when a hot feature needed to be fast tracked into production. The automated tests may not have covered the new feature but they gave a sound analysis of the effect (if any) of the new code on all existing features.

The package of automated tests focused on the FeedHenry platform front end. The tool employed to create and execute the tests was Selenium. It’s use on this project is described in the next section.

As mentioned previously, the testing performed for apps on devices was carried out manually on physical hand-sets or on device simulators. A test app that could be deployed to a range of handsets was used to verify an array of API calls and other ‘on-device’ features. An analysis of certain tools that would provide access to a wide range of devices, along with automated test facilities, was undertaken. Offerings from Perfecto Mobile [14] and Device Anywhere [15] were trialled and examined. However, the costs involved with using these products out-weighed any expected benefits and process improvements that could be achieved for this project.

5.2 Recording and executing the tests

![Diagram of test process]

Figure 2: The procedure for recording and executing the automated tests.
As new features were created, time was allocated during the test cycle to create new automated tests that would verify their functionality and stability through future releases.

The automated tests were created using the Selenium IDE [16]. This is an add-on available for the Firefox browser. A relatively simple to use tool, the tests were recorded, played back, edited and then saved to the Windows test server.

A recurring task was created on the test server in the Windows task scheduler. This would kick off the Selenium Remote Control (RC) tests at 5am every day.

- Selenium RC is a java based server which launches and kills browsers, interprets and runs the Selenese commands passed from the test program, and acts as an HTTP proxy, intercepting and verifying HTTP messages passed between the browser and the AUT.
- It can be used to execute tests using a range of browsers including Firefox, IE and Safari.

The selenium batch job was created to perform 3 functions:

- Execute the full suite of test cases across the specified browsers
- Write the results to disk on the test server
- Email the results to the appropriate team members

6 IPR Transfer

A very important aspect of creating a commercial company from a research project is the management of the IPR (intellectual property rights). A well governed software process that is maintained throughout the full life cycle of a research project can save enormous amounts of time during the formal IPR transfer of that project to a new company. In particular, the process can help ensure that the IP (i.e. code base) of the new company is maintained in a structured and clear fashion so that the following can quickly and easily be identified:

- Code developed and wholly owned by the new company which may/not be clearly identifiable (i.e. consistent file headers with copyright statements)
- Third party code used by the project, unmodified, which may/not be subject to specific license terms
- Third party code used by the project, modified, where the base code is not owned by the company but the modifications are

If the above items are not easily identifiable, the IPR transfer process could potentially incur very significant delays and elevated costs.

7 Conclusions

The TSSG engages in basic and applied research along with innovative commercial projects. It follows a model that tries to grow research and innovation outputs into tangible benefits for the community, e.g. job creation. The FeedHenry project has proven that this vision of a novel idea or a spark of inspiration being transformed into a viable commercial entity is valid and real.

The case study described in this paper outlines how engaging in software process, specifically agile, from an early stage is a sound strategy. Adapting and evolving the process as the project matures is a necessity to meet the quality and feature delivery expectations at any point in the life of that project.
Having a comprehensive and reliable automated test system is essential for providing the confidence to deploy new features on a regular basis. In conjunction with a V&V team that will run the required manual test spirals and police the agreed process, the combination is a recipe for success.

FeedHenry continues to grow and is now regarded as one of the leading ICT start-ups in Ireland. The award winning company continues to attract new customers globally in a wide range of sectors including telecoms, healthcare, retail, financial services as well as ISV’s and developers [17]. The Feed-Henry journey is a story worth studying and one that is destined to continue and grow.

8 Literature


[8] LaForge A; “Chrome Release Cycle”; https://docs.google.com/present/view?id=dg63dpc6_4d7vkk6ch&pli=1, 2011


[10] Bugzilla (http://www.bugzilla.org/)


[12] Redmine (http://www.redmine.org/)


[16] Selenium (http://seleniumhq.org/)

[17] BloombergBusinessweek

Author CVs

Phelim Dowling

Phelim has 15 years experience in the IT industry working in a variety of roles and companies. Starting his career with Intel he rose to the position of technical/project lead at their R&D facility in Oregon. Here he was responsible for coordinating and executing the development, testing and deployment of numerous mission critical products into Intel’s manufacturing sites worldwide.
On moving back to Waterford he spent time as a Test Engineer with Waterford Technologies. Following this he worked as a consultant project manager specialising in IT project delivery in the highly regulated pharmaceutical industry. In 2006 Phelim joined the TSSG Verification and Validation team and has facilitated on projects such as iServe, Muzu, FeedHenry and more recently the EU FP7 funded OpenLab initiative.