Case study - Team Foundation Server and Agile

Communications giant increases software development productivity, customer transparency and builds team spirit with Agile SCRUM and Team Foundation Server (TFS).

Introduction

Dunstan Thomas Application Lifecycle Management has been working with the Mobile Data Solutions division within a leading communications and infrastructure Company's Government, Mobile and Enterprise business unit, for over two years. During this period the Mobile Data Solutions' software development team have been rolling out software for leading UK police forces' Mobile Data Terminals (MDTs) and Personal Digital Assistants (PDAs). The division also provides software solutions to the ambulance, fire and coastguard services.

Key Points

- The Company used Agile SCRUM to put software development disciplines in place which delivered more mission-critical software improvements with less overhead while engendering a more helpful work ethic and better team spirit
- Indicative productivity has more than doubled in the team by using Agile SCRUM and TFS
- TFS has helped manage many aspects of the process, most notably monitoring and reporting of software development progress end-to-end
- Dunstan Thomas Application Lifecycle
- Management acted as a 'one stop shop' for all of The Company's needs from initial sounding board to Agile SCRUM training and consultancy, TFS provider and consultant

The Company secured a contract with a leading UK police force to supply their own Police Mobile Data Solution (PMDS) to the whole of the forces' MDT and PDA estate following an extensive trial in 2007. Using a combination of bearers, including GPRS and Airwave, the PMDS now gives more than 2,000 police officers secure wireless access to individual constabulary databases as well as national resources such as the Police National Computer, Voters, Emerald Warrants, National Mobile Phone Register and Gazetteer. Using the PMDS it is even possible for the police to use their PDAs to complete key administrative tasks such as issuing Fixed Penalty Notices while out in the field.

Such has been the success of this system in terms of keeping officers on the street, rather than behind their desks, that this force is currently extending the system to an additional 1,000 police officers. In addition, several other police forces are trialling the PMDS solution. Over the last two years The Company's software engineering team have been working on functionality and usability enhancements to the interfaces on these mobile devices as the Windows-based solution goes through User Acceptance Testing and roll out.

Waterfall provided a constraint

A new Software Engineering Manager joined to head up the software engineering team at Mobile Data Solutions back in late 2006 just before the Company started a 150 PDA and 20 MDT trial at another police force.

The Software Engineering Manager explained: *"At this stage, we were using the Waterfall software development methodology. This imposes the need for all functional requirements to be fully captured and signed off before moving onto the next phase, i.e. design and development. This was not working well for us because of the business pressure to ensure that all resources in the lifecycle were being fully utilised. As a consequence the decision was made by the business to start the design, development and testing phases in parallel prior to the proceeding phase completing. Basically, Waterfall was proving too much of a constraint for the team as demands grew to show progress on the project plan."*

"We were reliant on more than a dozen contractors for the delivery of the project. This created some additional risk in a project which was already very complex. We needed to take more control of delivering individual features so that the customer could see tangible progress in the form of a working product rather than time spent on the plan"

Before changing the development approach 15 per cent of test code was being returned to development as not fit for purpose. Much of this code had to be rewritten. As a result productivity was severely constrained , in both test and development phases, at a time when the team was completing the core development of the PMDS system and moving into a roll-out phase across several forces and public services.

The Software Engineering Manager added:

"It was also clear that if we were going to continuously support, improve and upgrade PMDS for our customer and other forces over time; we were going to need to increase our productivity. We needed to look at a more appropriate structure, methodology and process to achieve this."

Change was hastened by the need to increase quality while maintaining competitive effectiveness. A new, more productive way of working needed to be found fairly urgently. They began an evaluation of the more popular software development methodologies including Agile SCRUM which the team had successfully piloted in a small project in May 2008.

Agile transition begins

During October 2008, a Software Development Manager was recruited to help select the right software development methodology and tools to increase the productivity and effectiveness of the in-house team. The Company had an in-house software development team, made up of testers, integration specialists and technical design and delivery specialists. Further investigation was instigated into Agile SCRUM as well as alternative approaches. To support this work, the Software Development Manager sought out an independent Application Lifecycle Management (ALM) partner who could provide an overview of relevant software development methodologies, tools, training and support.

The Software Development Manager explained:

"The Company was looking for a 'sounding board' which was attuned to the Microsoft environment but which was essentially tools agnostic. We didn't want to be railroaded into selecting a specific methodology or tool. Dunstan Thomas Application Lifecycle Management proved to be the right fit for us because of a combination of its Microsoft Gold status, clear depth of knowledge and expertise in ALM, as well as its agnostic approach to tools selection."

Dunstan Thomas' Agile knowledge and hands-on experience of TFS proved very valuable in securing support for Agile from across the team. "Dunstan Thomas' consultants showed us how Agile had been applied successfully inside other Dunstan Thomas customers. We were able to paint a vision for what Agile SCRUM should deliver and use this vision to get the team to rally behind it. Dunstan Thomas also provided key tools and training to help implement it."

Source control

The first area the software development team worked on was source control. They worked hard to define their Branching Strategy to help speed up delivery of changes to live systems in parallel with making systems changes for other customers which may be at different stages of development. Essentially branching is about minimising risk. It is also about tracking the status of changes made in the source code of one branch that are need to be replicated in other branches, thereby minimising the risk of regression while resolving known issues.

The Software Development Manager knew that if the Branching Strategy could be set up and applied properly it would significantly reduce the risk of missing a customer deadline through regression. But it was not possible to do this with the team's incumbent software development tool which did not integrate effectively into their quality control tool HP's Quality Centre (QC). These constraints, together with poor productivity in the past using the tool, were the key drivers for a new approach in consultation with Dunstan Thomas.

TFS selection

In early 2009, the Company asked Dunstan Thomas to identify the most appropriate tools for their needs. Dunstan Thomas recommended looking at Microsoft's Team Foundation Server (TFS) which it had already used for other clients.

The Software Development Manager explained the natural link from TFS to Agile:

"Once we had gained team-wide agreement on TFS we were then able to quickly roll out Agile SCRUM. It fitted well with what we were trying to do - engendering greater team spirit and enabling us to work more dynamically and transparently with our customers on changes and enhancements."

TFS rollout

A Proof of Concept for the adoption of TFS was signed off in May 2010 which gave the team approval to proceed and purchase TFS licenses, together with Dunstan Thomas ALM TFS and Agile SCRUM consultancy and training support.

Phase I work began almost immediately with preparation for the migration of all source code into TFS which was completed in October 2010. Timing had to be carefully considered to ensure that there was no impact to the current deliveries in progress.
Phase II included migrating 'defect tracking' from QC to TFS. Process mapping and reporting systems were also finalised and rolled out by the end of 2010.

The Software Development Manager said:

"Moving QC to TFS definitely increased development throughput and improved traceability around defects and change requests in particular. TFS also enabled us to customise our workflows and then build automatically. It gave us early visibility of any build breaks which, to be honest, have been minimal since we moved to TFS."

Agile rollout

Agile was rolled out in the MDT (Mobile Data Terminals) team first. They responded immediately to the idea that Agile would minimise 'work in progress' by delivering running, tested, features, and help reduce 'technical debt'. Technical debt is a common problem in high pressure software development teams where developers find themselves having to take short-cuts in code writing to deliver functionality quickly, knowing that some code is likely to create future problems in terms of support and maintenance. This technical debt feeds through to new versions of the software which then become more expensive to maintain and resolve. Sprints are a key part of the Agile SCRUM process.

Essentially they are pre-defined sets of software development and delivery tasks which flow out of the requirements gathering process. A sprint defines precisely what coding jobs are needed and even puts a complexity and size value, called 'pointage', on each piece of code.

"The more sprints we do, the more history we have for the purposes of analysis and expectation setting. In other words if in the average sprint we are delivering equals a 100 points of code it is very unlikely that a sprint worth 200 points will be deliverable in the same timeframe unless new staff are added to the team. This approach gives us the ability to discuss delivery timescales realistically with the customer. We now know what 'velocity' to expect from each sprint based on past history of points cleared by a specific team. Being able to link changes in the source code to whoever wrote those changes, and what was actually done, is very useful in aiding collaboration between the development and test team members -a key tenet of Agile. Agile was great for us as it got us all focused on delivering fully functional and error-free pieces of code."

"Agile also encourages group working. Daily 'stand-up' meetings are carried out where developers explain what code they are building and what problems they are grappling with that day. This has led to better sharing of ideas and quicker solving of problems by getting the whole team thinking about a potential solution. Stand-ups also make sure that the whole team appreciates the mutual impacts of their own work."

The Management Team have increased the transparency around everyone's workload by having a white board in the room which is updated as things change using Post-It Notes. They also hold regular 'sprint reviews' to check how many points have cleared successfully and what has not gone through, to make sure the delivery remains on track. The team attended Dunstan Thomas-arranged SCRUM Master and Product Owner training which equipped them for the role of managing an Agile team through what was a fairly significant transition for all staff. Dunstan Thomas also provided training on Reporting Services for TFS to ensure the business could deliver reports when changes went live. Furthermore, to speed up productivity, the team decided to have training from Dunstan Thomas to fill any remaining gaps in knowledge about TFS' usage and on the use of Sparx's Enterprise Architect (EA) which it is currently evaluating.

"TFS can integrate with EA very well but TFS is also building its own UML modelling which may reduce the need to use EA for accurate capturing requirements. There is a constant need to re-evaluate potential tools to ensure they are still appropriate"

Core results from TFS and Agile

One of the major benefits of Agile, combined with TFS, has been a reduction in the number of returns to development of their delivered code which used to average nearly 15 per cent before and is now close to zero, i.e. 0-5. They also wanted to use the disciplines imposed by Agile, and tools provided within TFS, to automate software builds which it wanted to carry out overnight to increase productivity further.

The Company have already implemented delta builds to ensure that any compilation errors or broken unit tests are identified as early as possible. TFS integrates with FX Cop which is a tool for monitoring standards within the build. FX Cop indicates which sections of code deviate from pre-agreed standards. The Company also uses Resharper to highlight deviations from the standards to the developer while they are coding. There is even the capability to automatically 'find and replace' within this tool, saving even more time. Other tools are helping to automate some of the software building activity and further productivity gains will be realised from automating much of its unit testing work.

Sprint finish

Benefits also flowed from integrating software development, testing and building of tools within TFS. The Software Development Manager explains another key discipline of Agile SCRUM: "We identify functional items that the team will deliver in the sprint which have shortened to seven days for software development and build and a further seven days to test and deliver. It is all highly collaborative.

"We do now try to involve the customer in reporting the progress of each sprint. Once they understood that completion of the sprint did not mean the release of new functionality, but instead staged progress on that functionality, it became a lot easier to talk to them about these sprints. There is a real risk that progress can be misunderstood when you're open about each sprint's progress. For example, just because an item is fixed from a development perspective does not necessarily mean that it works. We were very careful to communicate not only the facts but also what that actually meant from a progress and risk perspective. We were also very careful about terminology. For example we refer to items in TFS that are ready for testing as 'Ready for Build' rather than 'Fixed'. We are now at the level where our customers get involved in the process and provide input into the content of each sprint. "

"In offering this degree of transparency, we have re-engineered the relationship we have with the customer. Previously we would discuss with them how long it is going to take to deliver this or that functionality. Then they would say 'please explain why it is going to take that long'. But the transparency which Agile and TFS gives the team more confidence that we are working with realistic expectations which we can stand by in conversations with the customer."

"We chose TFS over other tools principally because we wanted to take a more holistic view – managing the whole lifecycle of software creation and implementation rather than just trying to improve source control. It is really important to do that thinking before selecting the tool otherwise you have to force fit your working practices to a tool which may not be ideal for you and your team. Another key reason in choosing TFS was a consequence of our current strategy to focusing on delivering all software within the Windows environments."

"Our customers trust us much more and challenge us much less today. Agile and TFS have helped us build a much more open and mature relationship with them. Our project plans are rarely challenged

and when they are we always have clear evidence to show the rationale behind the scheduled activity."

Team building

Cohesion in the team is better as a result of Agile and TFS. The Company's software development team takes a lot more ownership in delivering working code to the test team. They tend to discuss in much more detail how they crafted the code and only then hand over to the test team once they have accepted the code in principle. The test team trust much more what they are receiving now.

"In essence Agile & TFS has helped engender a much more collaborative approach which gets everyone owning the software delivery and ensuring nothing, within reason, gets overseen."

Future plans

The Company is currently developing unit test scripts with a view to rolling out reliable automated testing early in 2011. Once they are fully satisfied with these scripts they hope to be moving towards a 'test-driven development' approach which will reduce re-work even further. They will also automate these 'unit tests' which will be run as part of the automated build process, using a unit testing tool built into TFS. In this approach inputs and outputs are defined and then are automatically checked every time a change is made to ensure these are still being delivered in the build and that no regression has taken place.

The final piece in the Agile jigsaw is to move to 'user stories'. Agile demands that in order to define a piece of functionality you put yourself in the shoes of a specific user and ensure that you know what they expect and need from a piece of functionality. It is hoped to adopt 'user stories' in the next customer project.

The Company is also looking to increase the collaboration between development and testing by getting feedback on software changes earlier. The aim is that once the team marks a feature as complete this initiates an automatic build overnight so that they are available for testing the very next day for them to test and feedback on. The idea is that they get software finished earlier in the sprint and leave a minimum amount of defect fixes to the end when they want to focus on closing up the sprint and reporting back to the customer in an orderly way before defining the next sprint.
In the near future, development, building and testing will all become a single smooth and seamless collaboration process in which delivering tested, running, features is everyone's focus and priority.

"This is the way an Agile sprint really should be – it means you deliver software daily – thus reducing remaining 'pointage' workload day-by-day as software is finalised."

"If new requirements are raised then they feed into the next sprint. A significant change of content of an in-progress sprint can result in that sprint being abandoned, along with any committed dates

until the impact can be assessed. Highlighting the cost of change to the client has meant that we have only ever had to abandon one sprint."

"Rolling out TFS and Agile has definitely resulted in higher levels of mutual respect, motivation and morale. Trusting the team to get on with the job, and providing the right development environment, has meant that team members have grown as individuals- taking a lot more ownership today in the quality of the software that they are delivering."

Why Dunstan Thomas

The Software Development Manager highlighted what Dunstan Thomas Application Lifecycle Management helped with:

"Dunstan Thomas' experience of TFS was a key differentiator. They are patient with us and very flexible when we have needed to delay training sessions or tool roll-outs. They also keep right up-to date on their ALM knowledge
which is very reassuring. "

"Dunstan Thomas Consulting's background has really stood out. The key thing is that they are tools agnostic so they really look hard at the whole market before selecting and building great expertise around a suite of best of breed ALM tools."

Dunstan Thomas Consulting
+44 (0) 23 9282 2254
www.dthomas-software.co.uk