

# CSM Practicing Certification Renewal Assessment

Name: Stephen Garnett

email: [steve.garnett@conchango.com](mailto:steve.garnett@conchango.com)

date: 25<sup>th</sup> October 2004

Scrum depends on the inspect and adapt mechanisms of process control to manage the complexity of projects. For inspection to work, everyone must know what is being made visible. To implement the Scrum process, such regulating mechanisms as defined roles, involvement versus commitment, time-boxes, and regular cycles are used.

**Describe one project on which you have used Scrum over the last twelve months.**

**Describe:**

**Purpose - what business goal was the project intended to deliver?**

The client is the largest and most respected vehicle management company in the UK, generating revenues in excess of £2bn. The company is in the business of remarketing cars: collecting, appraising, inspecting, valeting, transporting, selling, auctioning and delivering cars to end buyers. In addition to these activities the client manages fleets of vehicles on behalf of customers.

The Group IT Strategy was to migrate to Microsoft .Net technologies and improve their software delivery capabilities. The project (Inventory Management System (IMS)) was the first step. The existing remarketing processes were diverse and it was perceived would benefit from rationalisation and simplification.

The purpose of the project therefore, was to implement the first migration of systems from IBM AS400 to Microsoft .Net technologies. The current Fleet Management System (FMS) was to be replaced by a new Inventory Management System (IMS).

Primary Objectives

- Deliver the IMS System
- Deliver improved business processes for the management of vehicles
- Build client skill sets in designing and building Microsoft .Net systems, so that they can implement further system migrations to .NET with less help, with the aim of becoming self sufficient

**Length - what was the duration of the project?**

The project consisted of a 6 week period of planning and scaling, 7 development sprints and a release sprint, a total of 9.5 months.

**Cost - what were the budgeted and actual costs?**

The costs of the project are confidential, however the original estimated budget for the project was met.

**Value - what were the projected benefits and actual (if measured) actual benefits?**

The anticipated benefits at the start of the project were:

- Improved efficiency in vehicle remarketing
- Improve the capacity of Remarketing and Client Services teams
- Scaleable, resilient architecture for future systems migration
- Provision of a technology base that can support business needs over the next 5 years
- Increased client IT capability in Microsoft .Net technologies

At the point of the solution going live the anticipated benefits were:

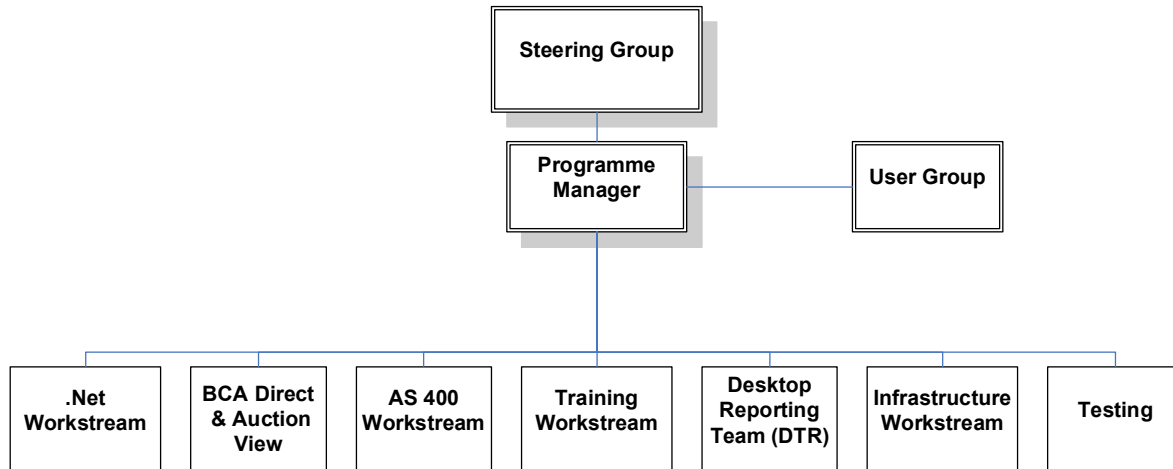
1. Closed and Open sales are now categorised enabling accurate reporting.
2. Sale Channel is now categorised enabling accurate reporting and removes the need for duplicate account numbers.
3. One source of data, i.e. all info comes from the branch system.
4. Better more accurate reports - Vehicle sales will process accurately (no data feeds), Stock days more accurate and match branch reports, Stock arriving on site are easier to pick up on reports (repo's and early terms).
5. E-sales channels vehicles can be loaded straight to website, no overnight feeds and amendments can be instantly applied.
6. E-sales now have a data verification process to allow more accurate first time sale uploads.
7. Get – task facility will drive down stock days, ensure vehicles are getting into sale channels, and highlight late delivers whilst also always giving operational admin staff a task to complete without the need for management intervention.
8. Ability to select sale allocations by 3+ or not solds will drive down stock days.
9. Ability to select transport movements by 3+ or not solds will help to remove slow selling stock from sites.
10. Postcode lookup will aid the speed and accuracy of transport movements.
11. Transport categories i.e. Compound, Auction etc will aid analysis of transport movements and give better cost of movement analysis.
12. Using Get Task functionality we will now have a real time view of overdue collections, appraisals and unallocated stock.
13. All data is now stored on the one database so negating the need to email, fax or phone through information. This information will be received in real time by both central departments and branches.
14. Real time data means increased accuracy of data.
15. Ability to allocate vehicles to sale and improve conversion rates.
16. Search criteria on the transport system will remove the need to go searching manually, using reports, to identify vehicles to be moved or collected.
17. We will become much more pro-active in managing not sold stock due to Get Task, Transport IBM search criteria and allocate to sale then we have ever been able to be in the past.
18. Standardisation of search criteria on all pages will make the system much easier and quicker to use in the fullness of time.
19. Ability to create transport and inspection requests within the same system improving the usability and understanding of the collect and inspect processes.
20. Due to having only one database reports can now be designed for both branch and central office use.
21. Biggest benefit to all is that you only need to use one vendor account number to cover all our services, i.e. Inspections, Electronic auctions.

The difference between the anticipated benefits at the beginning and the end of the project are and indication of the amount of business involvement and buy-in the project experienced due to the scrum approach. There were other sizeable projects being run within the corporation at the same time, none of which had as much engagement, commitment or buy-in from the business stakeholders.

**Size - how many people were on the project team(s), how were they organized into teams?**

The IMS programme consisted of a team of around 20 people which included training, change management, user testing, and mainframe resources. The core part of the programme was the .NET team which consisted of 4 highly experienced .NET developers, 4 novice .NET developers, a tester and scrum master.

The .NET team used scrum with elements of XP development practices. The other “workstreams” worked around and provided services to .NET scrum team.



**Teams - were the teams cross-functional and self-organizing? Were the teams collocated in an open space? Were the teams physically separated within one location, or located in more than one physical location?**

The scrum team itself (the pigs), were collocated in a war-room environment with wireless connectivity and 4 whiteboards. As the sprints progressed the team was added to as required for the specialist skills needed for the sprint e.g. creative designer, infrastructure designer, integration specialist.

At the start of the project the team was structured by specific role, but as the team grew over the sprints they became self-organising.

**Initiation - how was the project initiated? How was the team trained to use the Scrum process?**

The project was initiated in a rather formal way. A project initiation document was produced with high level functional and technical design. These products were used to create an initial estimate of effort and costs which was formally presented to and agreed by the board.

From this point onwards 7 development sprints were conducted and a release sprint.

We held a programme kick-off meeting on the first day of sprint 1 identifying approach, structure and the key elements of scrum, such as 4 week sprints, sprint planning meetings, user presentations, and sprint review meetings.

Members of the team were progressively sent on the Certified Scrum Masters course during the project, and as the team progressed they became more agile.

### **Reporting - how did you report progress to management and the customers?**

Throughout the project I produced a weekly status report that became extremely anemic by the 5<sup>th</sup> sprint onwards.

Because of the huge buy-in to the approach following the initial sprints, the main reporting period became the sprint itself. The steering group meetings were held after the user presentation to review progress against initial estimates, review and prioritise the product backlog, and report on budgeting.

### **Change - what difficulties were surfaced by Scrum that had to be resolved? How were these resolved?**

The most significant issue was that of performance. The great thing about Scrum was that the issue was raised at the end of the first sprint. A piece of 3<sup>rd</sup> party integration software was performing poorly and we were able to call on them for resolution and support at a very early stage.

The key thing is that Scrum doesn't solve or negate the problems faced in software development, rather it makes the issues visible immediately to everyone. This focuses the team on resolution, and gives the business plenty of time to make to best decision possible and execute corrective measures.

### **Management - what was the previous role of the ScrumMaster? Who took on the role of Product Owner? To what degree were they successful in fulfilling their roles?**

Previous to being the scrum master I was a project manager. It was difficult to get used to the change in culture and structure that scrum introduces. I felt there was a general dispersal of power and responsibility from myself to the team as a whole. This is much better, as the team became more proactive in getting work done, identifying and solving problems, and taking ownership for the solution.

By separating out the traditional Iron Triangle so that the Product Owner is responsible for the scope of the product, the team as a whole responsible for delivery, and time-boxed delivery, scrum enables everyone to drive out the delivery and the scrum master becomes the facilitator, obstacle remover, and service provider to the team.

### **Engineering - what software engineering practices or environment had to be changed?**

Plenty... we started sprint 1 with a basic environment with daily automated builds. During the project the team decided to implement the following practices:

- Automated Unit Testing
- Continuous Integration
- Refactoring
- Peer Design and Review
- Paired Programming

The decision to adopt increasingly higher level of development discipline lay with the team. Usually, during the sprint review meeting the team would identify something that didn't go well or could have been done better. The identified improvement with associated estimate would move into the sprint planning meeting for the next sprint. E.g. adopt continuous integration 16 hours.

**Stabilization - for how long did the software have to be stabilized before it could be released? How did you structure this stabilization process?**

The release sprint (4 weeks) covered the stabilization of the solution. This included prioritizing any bugs, conducting load, regression and stress testing on a fully operational UAT environment, that had been established on earlier sprints.

**Success - to what degree was the project successful? To what degree was the Scrum process instrumental in the success of the project?**

The scrum process was extremely effective in building stakeholder buy-in and involvement on the project.

The incremental deliveries enabled everyone to be clear about what was being produced and helped the decision making processes on scope and changing requirements much easier.

The morale and enthusiasm of the team remained positive throughout the project, and the productivity levels were much more consistent than traditional projects where there is often a lull in activity through analysis, design and the start of development phases.

**Scrum Process - to what degree was the Scrum process implemented "out of the box?" To what degree did you have to modify the Scrum process for this project? For each modification, how did you formulate the modification so that the basic inspect/adapt mechanisms continued to function? What parts of Scrum couldn't be implemented, or failed, and why?**

The scrum process was fully implemented from sprint 3 of the project.

Sprint 1 had scrum meetings, a user presentation and sprint review meeting.

Sprint 2 had the addition of the sprint planning meeting with the entire team involved, rather than just the project manager and lead developer.

Sprint 3 had a formal product owner and backlog. Up to that point we had not been using a formal product backlog with product owner. Rather we had a high level requirements list that was not prioritized by the business and did not have a formal product owner.

From sprint 4 onwards all the facets of scrum were being used.

**How do you cause the accuracy of Product Backlog estimates to improve? To what degree does their accuracy matter?**

The accuracy of the product backlog can be improved by reviewing the effectiveness of estimates based on team performance during a sprint, and spending more time designing the solution to be able to more granularly define tasks. However, the need for more granular and accurate estimates needs to be considered within the overall objectives and momentum of the project.

If accurate estimates are required in order to help the project move more effectively to conclusion then they should be carried out. However if, for example, a 15% inaccuracy is acceptable within the project governance then spending more time refining these estimates is a waste of valuable development time.

**How do you cause the accuracy of what a team commits to for a Sprint to what the team actually delivers?**

Well the first answer obviously involves the estimation area, but a more important consideration is the definition of "done". The team at the beginning of a project are a group of individuals with varying levels of skill and discipline, and different perceptions of what is a completed piece of software.

Key questions to ask are around development practices such as commenting, automated testing, use of continuous integration, peer review, or refactoring amongst others.

The aim of the team should be to steadily increase their practices as a team, to improve quality, and to be at a point where everyone shares the same perspective of when a piece of work is done.

Other factors that may cause a team to not reach what they have committed to include scope creep, resources being pulled onto non-project work, or obstacles not being removed by the scrum master efficiently.

On this project we had a problem in sprint 2 where we thought there were 5 scenarios to a use case, but there turned out to be 13. Instead of communicating this issue immediately and de-scoping we tried to develop all 13 which led to quality issues, and long hours which tired the team. The positive side is that we all openly discussed this issue in the sprint review and we did not make the same mistake again.

If a team consistently under achieves on its commitments, then there is an underlying issue that may require corrective action such as more detailed analysis up-front or development coaching.

**What metrics do you use to track the development process? Which metrics have been changed, removed, or newly implemented as a result of using Scrum?**

The best way to measure progress is to play with the software released each sprint, and review the bug list.

Beyond this, the key metrics I look out for now are:

Unit test coverage

Unit test success rates

Comment to code ratio

Number of builds per day (continuous integration)

Sprint burndown

Project burndown

**5. What type of training, resources, or tools would best help you successfully employ Scrum in the future?**

1. Whiteboard and pens
2. Project Room
3. Source Control
4. n or j Unit for unit testing

Anything on top of these is a bonus, but these are fundamental.

**6. (Optional) Scrum and Extreme Programming are sometimes used together. What must be considered when this is done?**

I would say that you need to separate out delivery process from development practices. i.e. use the scrum 30 calendar day sprint, with backlogs, planning meeting, user presentation and review meeting for the delivery process. Then for development practices adopt the XP principles.