Scrum Notes

Introduction

The following words and pictures show a Scrum team in action at a Design Agency.

The project was geared to assembling a team to present a bid for a global teleco. This involved teaching the team how agile works and providing them with confidence to carry out 2 hour one-on-one practical sessions with the client organisation.

The project was very short – 2 weeks. We compressed the sprints to a daily basis.

The examples of various team activities provides an insight into the softer side of Agile Development that I facilitate and beleive put people first. There is also a picture of a Scrum Wall for the technically minded.

Meet the team

We had a culturally diverse team preparing a bid for a multinational telecoms client. The team consisted of: 2 Product Owners, 2 Scrum Masters, 2 QAs, 2 Testers and 4 Developers.

This is us having a “chatacino”. It was fun toget out of the office and have the occasional extended stand up at the coffee shop.

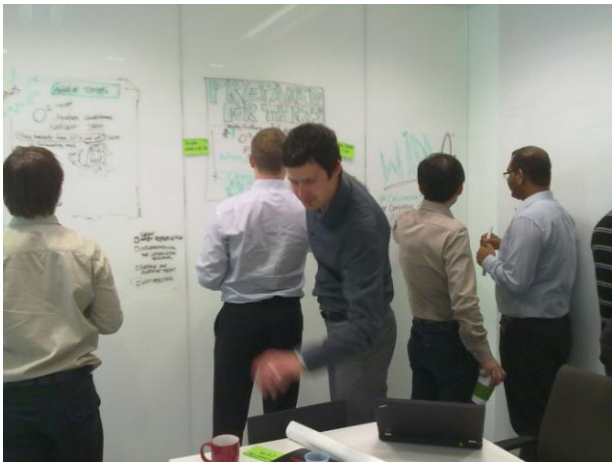
The following sections will look at planning, communication, measurement and retrospectives. Finally a set of lessons learned will be presented.

Sprint Planning and Shared Mental Models

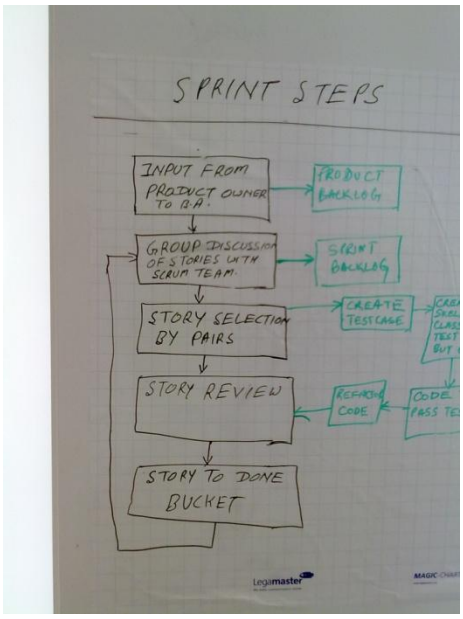
Planning poker was carried out using bunches of fives since we were in a short duration project.

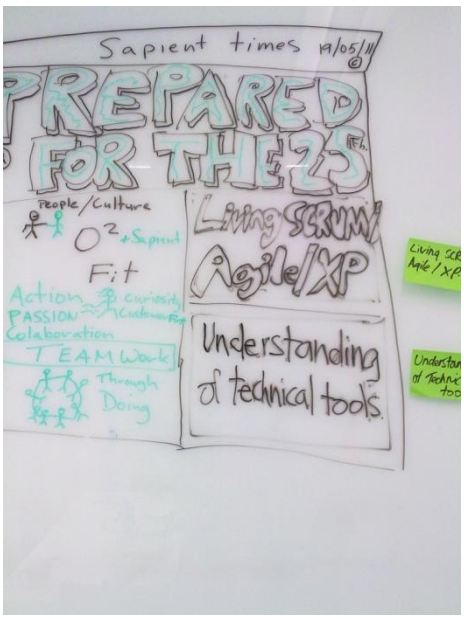
I joined the project with most of the team already in place. On day 1 I was acquainted with the processes of the company and quietly observed what was happening in the team. I recognised that the team was disjointed and that there were opposing view points.

To ascertain the different viewpoints in the team I facilitated a Magazine Cover session.



The results below show the most different productions. The one on the left was produced by a pair of developers and the one on the right from the Product Owners. Can you guess which version we used to focus our shared mental models on?





With the team now aware that there were varied viewpoints of what we were trying to achieve we had a group vote in what aspects of the process they wanted to improve. This was done by starting with 11 dimensions one would expect in an innovative team then extended the list to include other peoples wishes. Prioritisation of aspects to address was achieved through point voting where members placed plus signs next to the topics they though most important.

The outcome in this instance was to have more pair work – but with different functional team members. I have been trained in being able to select the appropriate type of team building and creative task from a toolbox of about 150 and then facilitating through the appropriate form of meeting.

Communication

The next day I facilitated a Blind Man Walking session with the functional pairs split. This was the turning point in the communication level in the team. Here is a PO being led by a developer.

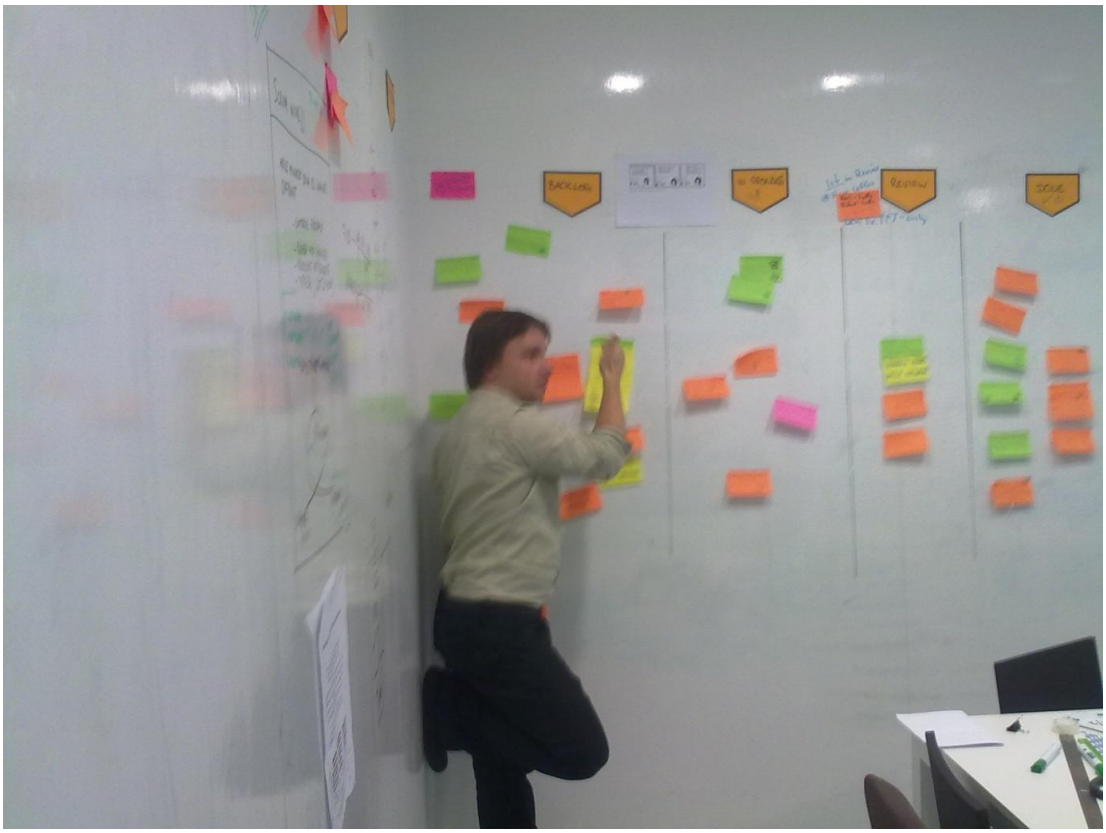
I must admit that not everyone enjoyed the ice breaker approach. Two members on the blind man walking went for coffee instead. It was like herding civets...

In general the closer the members were to the business (Product Owners and QAs) the more they enjoyed the sessions. The developers found them abstract and did not enjoy being moved out of their comfort zones. (See lessons learned)

The dividends of the sessions were reaped during the code dojos. The increased level of communication made the more hesitant and introspective team members realise that they could contribute to the debates that developed during the dojo. (I do not have pictures of the code-dojo as I was using my phone as a stop watch rather than a camera)

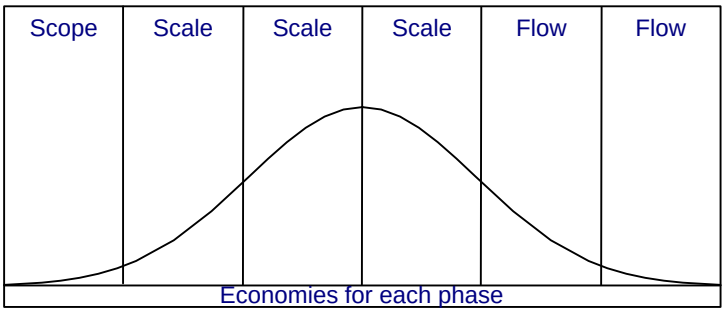
Measurement and visibility

All teams need to be able to see the progress – and importantly so does the business. We used a Scrum wall that was updated daily to show tasks moving from the backlog through the phases of until they were done-done.



My fellow scrum master, Dieter, running the board. He views Scrum as a CMMI Level 3 compliant process. German efficiency in action. I like his approach for planning and reportng. Our teamwork was of a high level and aim to recreate it whenever possible.

Now for the abstract bit: Selecting the appropriate agile method at the correct phase of the project is useful to realise that corresponding efficiences of scope, scale and flow. (Ghoshall, Seddon)

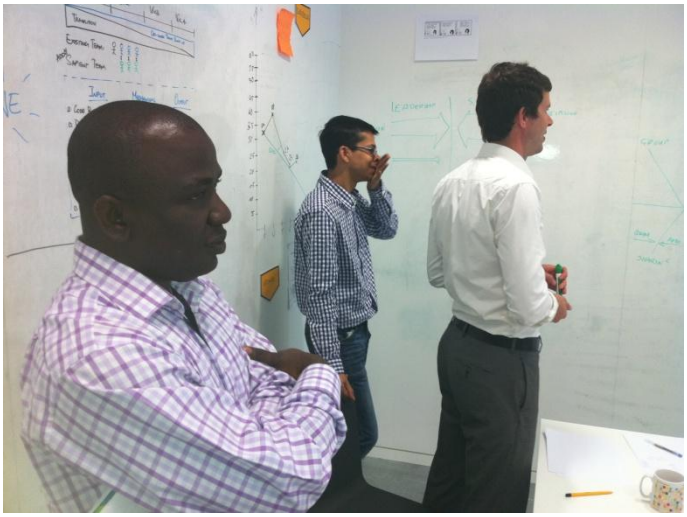


The generic diagram above shows how to use different methods at the different phases in a project. Scope is more discussion and spike activities to select technologies and create shared mental models – it could be used to refine release planning. Scale is best achieved in sprint releases and Flow can be realsied using Kanban – or a helpdesk type ticketing system and a knowledge base.

Retrospectives

Each sprint should have a retrospective to generate and share lessons learned. I have seen retrospectives range from 1 hour to 1 day in duration. 1 day retrospectives are more like death by meeting.

Here is a retrospective I facilitated in action. The teams produced force field and fishbone diagrams to create contrasting viewpoints of various topics that had cropped up during the sprint. It was the first time that the majority of the team had used such diagrams. This focused the dialogue and debate with warm actions rather than plodding through the same stuffy agenda and minutes processes. They enjoyed the experience.



The retrospective led to another block vote to decide on the topics to address.

This is an example of a self organising team using double loop learning to alter its environment. This powerful technique has benefits in terms of morale and productivity. It is also very cool and subtle.

Lessons learned

1. Not everybody enjoys being moved out of their comfort zones. Leadership requires knowing how far individuals can be pushed and when and why to do it.
2. Ice breaker activities lead to better communication, usually.
3. Code dojos transfer know-how and are enjoyable.
4. The current Agile methodologies are not ideal. No model is. That said they provide understandable frameworks that allow measurement and visibility across the different stakeholders.
5. The realisation of self-organisation is best achieved through double loop learning where the team can alter its environment by changing variables it controls but may not know exist.