

Agile Planning with a Multi-customer, Multi-project, Multi-discipline Team

Karl Scotland

Development Team Leader, BBC Interactive
karl.scotland@bbc.co.uk

Abstract. Most XP literature refers to teams which work on a single project, over a number of months, for a single customer using a narrow range of technical disciplines. This paper describes the agile planning techniques used by a team which works on multiple projects, for multiple customers, using a wide range of multiple disciplines. The techniques described were inspired by the agile practices of XP and Scrum. A small case study of a project shows how the team is able to collaborate with their customers to deliver maximum value under tight conditions.

Keywords

Agile, XP, Scrum, Planning, Releases, Iterations, Stories, Wiki, Interactive TV.

Introduction

The development team is a small group, currently seven strong, developing enhanced TV (eTV) services for the BBC. The range of deliverables includes content entry tools, content transformation processes, and Set Top Boxes (STB) applications. Services are required to be compatible for broadcast in the UK on the digital satellite (DSat) platform, and the digital terrestrial (DTT) platform.

Projects

The team works on a large number of very short projects typically of a couple of weeks in length. This means that the workload is unavoidably chaotic. The short timeframe of each project means priorities must be robustly managed both within projects and across them. In addition, there is little opportunity to get into a rhythm during a project. The solution is to treat all projects as a single ongoing project.

A further complication is that the team works in a domain in which they have little control over milestone dates. Some projects have absolutely immovable delivery dates, such as a major sporting event like Wimbledon, while other projects have unknown delivery dates, due to the nature of channel scheduling. In either case, they must always be in a position to deliver something which is acceptable to broadcast. DSat services are also required to be tested by the platform operator. Testing slots are limited, and must be booked eight weeks in advance, adding a further constraint to planning.

Customers

The large number of projects worked on naturally means that the team deals with a variety of different customers. The majority of these customers are internal producers; the people with editorial responsibility for creating services. They are currently located on the same floor as the technical team, and are hence always easily accessible. Graphics designers and operational support engineers, and the technical team itself are also considered to be customers on occasions, with requests for particular design and navigation, deployment or strategic implementations. To manage the numerous producers all vying for technical resource, a creative director is responsible for prioritizing work, and ensuring that the producers speak with one voice. This is a role comparable to the King in Ron Jeffries' *Petition the King* metaphor [1].

Increasingly, the team is working with producers from other divisions within the BBC, located at other sites. In order to manage the decrease in accessibility, an internal producer is usually appointed to be a 'proxy', representing the main producer, and to provide a local point of contact. The main producer is also encouraged to be on-site as often as is necessary to keep the communications and feedback as high as possible, and to enable them to learn the complexities of the domain where appropriate.

Disciplines

The interactive TV domain requires the team to use a wide range of skills. These are predominantly:

- OpenTV – an object oriented C based API used for DSat
- MHEG5 – best described as a cross between a markup language and assembly language used for DTT
- Perl – used for content processing and transformation.
- HTML and related web languages – used for simple content entry and publishing tools.
- C++ - used for more complex content entry and publishing tools.

This diverse range of skills, some of which are very specialized, means that maximizing the teams productivity can be challenging. There is a danger of being too dependant on gurus, causing a bottleneck in development and a truck-factor^[1] of one. As well as using agile planning techniques to manage this, other XP [2] practices such as pair programming and open workspace are used in order to be able to deliver flexible amounts of software across the different disciplines.

Release Planning

Release planning happens three or four times a year as the culmination of a commissioning round. This is a process through which the various divisions of the BBC can submit proposals for services they would like the team to build. Over the course of a day at most, these proposals are whittled down into a simple, high level, long term plan, which is put together in Excel. This plan allows the blocking out of weeks of work to different projects based on the project proposal, without worrying too much about the fine details of the projects.

In order to help make the high level plan as accurate as possible, without any detailed information, the work is split into three streams, based on a stream for each pair, and projects are split into the different disciplines; DTT, DSat and Content. The high level plan is also a useful place to make a note of any key dates, such a transmission, or test, along with other small pieces of work.

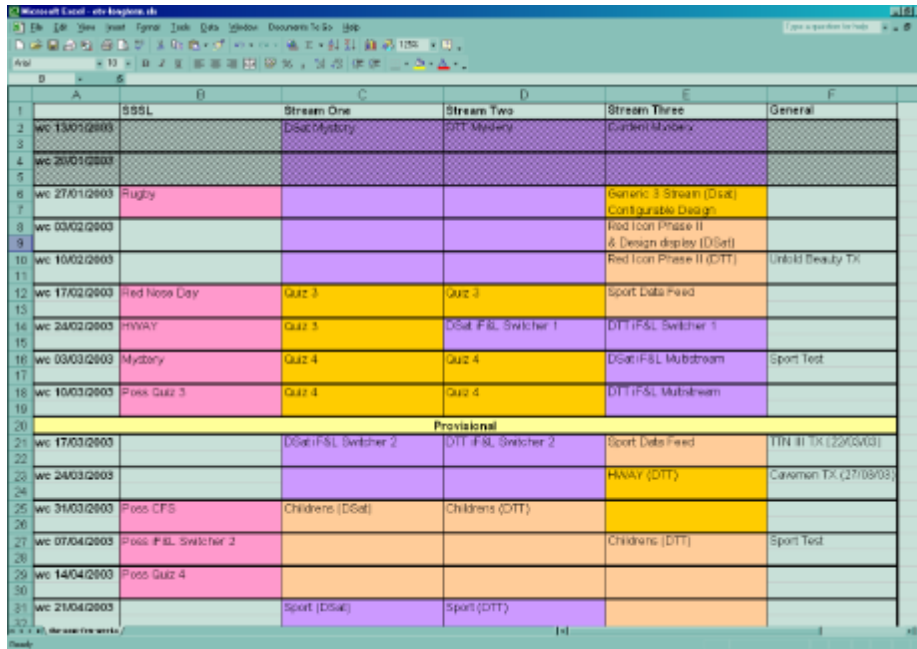


Fig. 1. Example Long Term (Release) Plan

The long term plan is reviewed against the actual progress at the start of weekly iterations, and a project's scope may be adjusted in order to keep on track, or the time allocated to a project may be adjusted in order to allow the necessary functionality to be completed.

Iteration Planning

A 'provisional' line on the long term plan marks the point up until which the team has planned projects in more detail. As each project approaches its planned build time it is broken down into stories, and each story is estimated using a simple scale of 1 to 5, representing its relative size. A 1 point story would be something small, simple and low risk, such as configuration work, and a 5 point story would be something large, difficult and high risk such as a use of complex new technology.

The team has tracked its velocity by recording the total number of points for stories completed during the weekly iterations, and charting in Excel the trend of the average points completed over the last 5 weeks. The velocity trend is used as a guide to how many stories can be planned into the available time.

While some on the XP mailing list [3] have recommended against charting a velocity average, as opposed to a strict use of Yesterday's Weather, the team's experience is that such short iterations lead to an unstable velocity, as shown in Fig. 2. While a review of stories and estimates should lead to an improvement in estimation and minimize this effect, the use of a trend provides the more stable indication of velocity which is essential for accurate future iteration planning, while still allowing the rapid feedback of one week iterations.

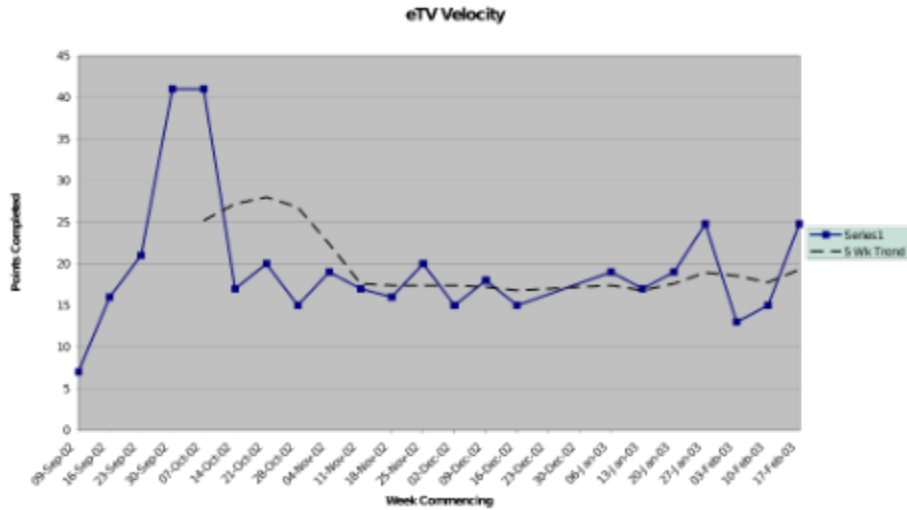


Fig. 2. Velocity Chart

Tracking

Once projects have been broken down into user stories, estimated by the technical team, and prioritized into iterations with the producers, their progress is tracked using a wiki [4]. A simple hierarchical structure is used, with an overall projects page, which links to page for each project, each of which links to a page for each discipline (DSat, DTT or Content), which finally link to a page for each story. Where stories are duplicated across the different disciplines, a common page is shared.

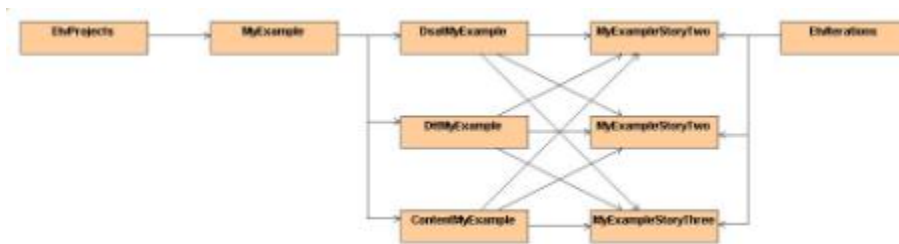


Fig. 3. Wiki structure

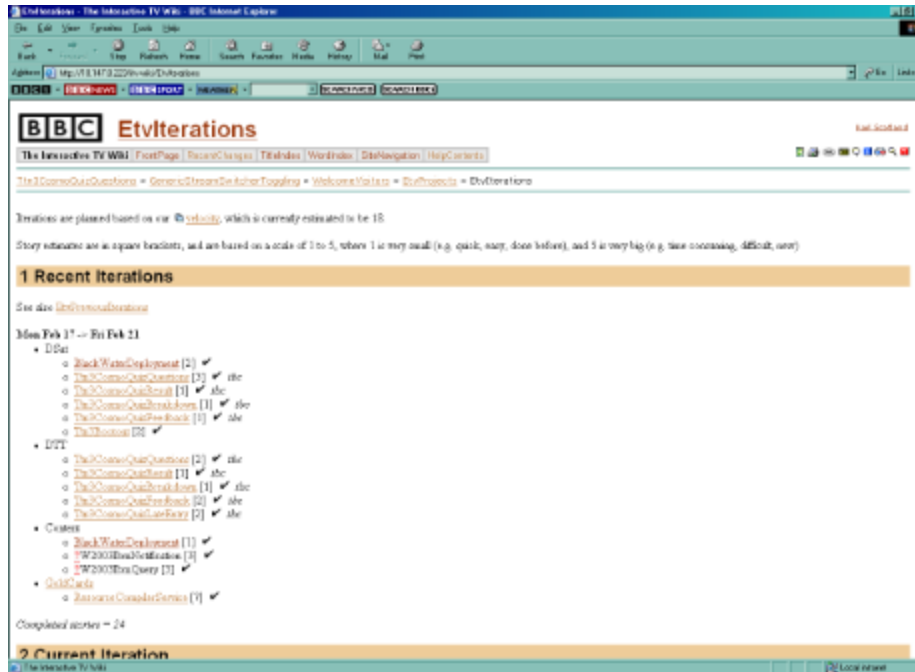


Fig. 4. Wiki Iterations Page

Figure 3 shows this structure for a project “My Example”. It also shows a separate page which is used for the iteration plan. This page, an example of which is in figure 4, groups links to the story pages according to which weekly iteration they have been planned into.

The wiki allows details and decisions about the projects and their stories to be easily documented. As the weeks progress, the various pages can be quickly updated to reflect current status, and they are easily available for viewing by the whole team.

Each week, that iteration’s story cards are also put up on the wall to highlight the current work, and to provide a focus for regular stand-up meetings. These are not always every day, in the same location, (and chairs are allowed), and are therefore not as strict as Scrum [5] advocates. However, they facilitate the necessary level of communication to track progress on a daily basis.

Example

A recent project had been allocated two streams for five weeks of work in the long term plan. The team’s velocity at the time was 17.5, so to simplify the numbers it was rounded up to 18, which meant that each of the three streams could represent six points of work per week. Therefore, two streams over five weeks represented 60 points. Over the course of a few days the editorial proposition was discussed with the producer, and stories were written and then estimated. When the actual planning began, the total number of stories was 114, adding up to 228 points, which was significantly over budget!

The first task was to cut down the proposition to include only those stories which were absolutely essential. In other words, if they weren’t included, the service would not be worth launching. This reduced the points total to around 120 – still double what we had time for.

The next step was to discuss the possibility of having extra time with the creative director. This resulted in a further two weeks of one stream i.e. 12 points, increasing the budget to 72.

At this point the situation seemed pretty hopeless, and it was obvious that something drastic was going to have to be done. After a short break for lunch, the producer was able to come up with a simplified proposition, for which new stories were quickly written and estimated. This resulted in a new total of 31 stories, totaling 63 points, giving some room to play with as the project progressed. It was decided that this was enough to begin development with, and progress would be monitored before deciding on possible extra stories.

Over the course of the project a number of further new stories came out, as functionality was completed and the producer reviewed and tested the developing service. Some of these were prioritized and implemented, while others could not be completed. Some of the original stories were also reprioritized and implemented. The final delivery consisted of 45 stories, totaling 84 points.

Table 1. Summary of project stories

	Implemented	Not Implemented
Original	35 (72 points)	79 (156 points)
New	10 (12 points)	5 (10 points)

While these numbers are inherently crude due to granularity of the story points, they give a good indication of how much the final project was reduced from the original aspiration. Within a very short space of time, the group was able to negotiate a plan which was satisfactory to everyone. It was realistic and achievable by the technical team, and met the editorial needs of the producer. In fact, the final service contained more functionality than originally planned, and the producer was extremely happy with the service. In an email to the team, the creative director said, "It's the best. It's absolutely fantastic!"

The process was not easy, (it was very intense), but the unanimous consensus was that it was better to go through the pain at the start of the project when it could be controlled and minimized rather than the end.

Conclusion

The team is able to manage the various complexities of multiple customers, projects and disciplines by working in a way which allows them to treat the work as if there were only a single customer, project and discipline. Thus they work on a single pool of stories which are shared by all the different customers, projects and disciplines.

The practices allow the editorial team to juggle all the projects, prioritizing and balancing the scope of each to maximize value for the business, while at the same time allowing individual producers the freedom to fine tune their propositions, either because of resource or time constraints or because they have changed their mind.

In general, the process is "barely sufficient" [6], and the plans are seen as "hypotheses to be tested rather than predictions to be realized" [7] so they are constantly adjusted to reflect reality.

Acknowledgements

Thanks to everyone involved in the eTV team at the BBC for being willing to adopt agile practices. Also, thanks to Humphrey Lau and everyone else who has reviewed this paper.

Biography

Karl has been developing Interactive TV services for the BBC since October 1999, and has led the eTV technical team since May 2002. He has been promoting agile methods since discovered XP in late 2000.

References

1. Beck, K.: Extreme Programming Explained. Addison-Wesley, 2000.
2. Jeffries, R.: Petition the King, <http://www.xprogramming.com/xpmag/PetitionTheKing.htm>
3. Extreme Programming Yahoo Group. <http://groups.yahoo.com/group/extremeprogramming/>
4. Leuf, B. and Cunningham, W.: The Wiki Way. Addison-Wesley, 2001.
5. Beedle, M. and Schwaber, K.: Agile Software Development with Scrum. Prentice Hall, 2001.
6. Cockburn, A.: Agile Software Development: Software Through People. Addison-Wesley, 2002.

7. Highsmith, J.: Agile Software Development Ecosystems. Addison-Wesley, 2002.

[1] The number of team members that have to hit a truck before the project is in crisis.