SCRUM based Enterprise Architecture Planning Process

by Charles Edwards. 1st April 2007.
Version 1.00

Abstract

SCRUM is an excellent simple, streamlined and proven mechanism for managing people in a collaborative team-working environment in an Adaptive manner (over a Predictive manner).

This paper first describes what SCRUM is in a generic context and then goes on to explore how it could be adapted for use by an Enterprise Architecture practice to manage its activities effectively.

Adaptive over Predictive Project Management

One fatal mistake made by people thinking about Agile methods is to assume it allows for chaos and random reaction to problems (or “fire-fighting”). After all if we are not working according to a plan, then we must be reacting to situations?

On the contrary, Agile methods are extremely disciplined and pro-active. They insist on planning, but not more than about a month into the future in any great detail, because we know change is going to happen, so why waste the time planning, only to re-plan later? Sure we can plan in the large in broad-brush strokes to assure the general direction, but that is more direction setting than detailed planning.

SCRUM – what is it?

This section describes SCRUM and how it works, but has been slightly modified to make the context less software development like and more generic for any team.

Background of SCRUM

Taken from wikipedia.org: “Scrum is an agile method for Project management. They noted that projects using small, cross-functional teams historically produce the best results, and likened these high-performing teams to the scrum formation in Rugby. In 1991, DeGrace and Stahl, in "Wicked Problems, Righteous Solutions" referred to this approach as Scrum. Jeff Sutherland and Ken jointly presented a paper describing Scrum at OOPSLA'96 in Austin, its first public appearance. Ken and Jeff collaborated during the following years to merge the above writings, their experiences, and industry best practices into what is now known as Scrum. Although Scrum has a theoretical basis in empirical (or evidence based) process control, its practices have all been empirically derived from extensive Scrum practice.

Its intended use is for management of software development projects, and it has been successfully used to "wrap" Extreme Programming and other development methodologies. **However, it can theoretically be applied to any context where a group of people need to work together to achieve a common goal** – such as setting up a small school, scientific research projects or planning a wedding.

Although Scrum was intended to be for management of software development projects, it can be used in running maintenance teams, or as a program management approach: *Scrum of Scrums*.

Characteristics of SCRUM

Five main characteristics ensure **a living backlog of prioritized work to be done**;

1. Completion of a largely fixed set of **backlog items (work item list) in a series of short iterations or *sprints***;
2. A **brief daily meeting or scrum**, at which progress is explained, upcoming work is described and impediments are raised.
3. A **brief planning session** in which the backlog items for the sprint will be defined. (once per sprint)
4. A **brief heartbeat retrospective**, at which all team members reflect about the past sprint. (once)
5. ***Scrum* is facilitated by a *ScrumMaster*, (Iteration planner)** whose primary job is to remove impediments to the ability of the team to deliver the sprint goal. The ScrumMaster is not the leader of the team (as the team are self-organising) but acts as a productivity buffer between the team and any destabilising influences.

Scrum enables the creation of self-organising teams by encouraging verbal communication across all team members and across all disciplines that are involved in the project.

A key principle of Scrum = Adaptive

A Key principle of Scrum is its recognition that fundamentally empirical challenges cannot be addressed successfully in a traditional "process control" manner. As such, Scrum adopts an empirical approach – accepting that the problem cannot be fully understood or defined, focusing instead on maximizing the team's ability to respond in an agile manner to emerging challenges.

Notably missing from Scrum is the "cookbook" approach (or Predictive approach) to project management exemplified in the Project Management Body of Knowledge or Prince2 – both of which have as their goal quality through application of a series of prescribed processes.

Adaptive Project Management

The following are typical practices of SCRUM (some of the wording has been modified to suit the context of Enterprise Architecture Practice):

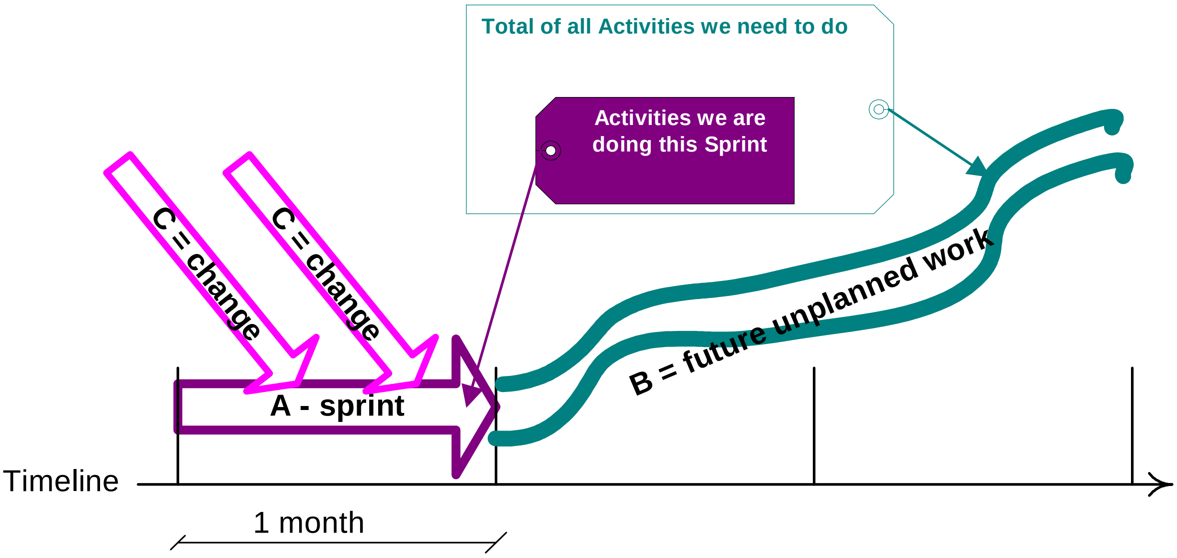
- **Customers** become a part of the team. (i.e., Customer must be genuinely interested in the output.) – Validate what you have planned to develop.
- **Frequent intermediate deliveries** with working functionality. – Incremental work and releases (may be internal only) – providing you an opportunity to validate and verify at shorter intervals rather than only at the end; thereby, providing you time to fix, and reducing the cost to fix.
- Frequent **risk and mitigation plans** developed by the team itself. – Risk Mitigation, Monitoring and Management (risk analysis) at every stage and done genuinely – Make it live, and continuous activity.
- **Daily status discussion** with the team. – Stand-up meetings (accomplishments, to be accomplished, issues / concerns / risks). Following three questions are asked of each person:
    - What have you done since yesterday?
    - What are you planning to do by tomorrow?
    - Do you have any problems preventing you from accomplishing your goal?
- **Transparency in planning** and module development – Let everyone know, who is accountable for what and by when.
- Frequent **stakeholder meetings to monitor progress** – Balanced (Delivery, Customer, Employee, Process) Dashboard updates – Stakeholders' update – You have to have Advance Warning Mechanism, i.e. visibility to potential slippage / deviation ahead of time.
- **No problems are swept under the carpet**. No one is penalized for recognizing or describing any unforeseen problem.
- Workplaces and **working hours must be energized**. – "Working more hours" does not necessarily mean "producing more output."

Suggestions for an Enterprise Architecture team Planning Process

The essence of an Adaptive planning environment

At the core of Adaptive planning is discipline. A cool, calm collectedness within a mad changing world.

- **A = For one month we try and keep as much as possible constant** – This allows us to make and show good progress in an orderly fashion. We also cater for expected change, by building this into the plan. E.g. A set chunk of time set aside for Project review requests.
- **B = Into the future we hold a very high level / general direction plan and backlog list** – no detail on the plan, it's just for alignment of all. We also keep a list of all backlog items still to do, which is actively managed.
- **C = Unexpected change that comes along** – We evaluate it and decide if we are going to change the plan or not. Most of the time it can wait at longest a month, if not something else has to give.


Total of all Activities we need to do
Activities we are doing this Sprint
C = change
C = change
A - sprint
B = future unplanned work
Timeline
1 month

Agile / Adaptive planning process

1. **A base plan for at most a month is defined**. This is done in a disciplined fashion driven by priority, risk, requirements, etc.
   - It is NOT a whimsical list of "to do's".
   - It is assigned activities and tasks of work to individuals out of the Work Activities backlog list. These items are then marked as being done in this sprint.
   - Team Individuals are involved in accepting and taking the work on.
   - Where collaboration takes place workshops and meetings are planned and booked a month in advance according to the plan.
   - The concept of time-boxing should also be used / considered in this context.
   - Also bear in mind that if we know we are going to get n projects that need some time for evaluation being presented the team during the sprint, we can plan for this and allocate time and resource to expect this as part of the plan up front (with some backup work if it does not materialise.)
2. **All other work is left** in the Work items list for future sprints.
3. **Any changes that come in during the month's sprint** - once the plan has started are then aligned to this sprint plan. (Expected to be unexpected)
   - *Most changes can wait until the next plan (sprint)* – 80% usual.
     - If we make the team environment constant for at least a month – then we will actually get somewhere and do things properly as planned.
     - Remember - at the end of this mostly constant sprint, everything is evaluated in the planning session for the next sprint and at that time all the most important items are brought into the plan.
     - One the one hand avoiding poor planning - mantra along the lines of – "Don't make your bad planning our problem".

  - On the other hand genuinely facilitating business change - without being critical of typical business change, such as new markets opening up, new opportunities arising, new information being discovered, etc.
- *Some important changes could be done in this sprint* - by taking something less important out. – 20%
  - Ideally we like to minimise these to VERY HIGH priority interrupts.
  - Not just to please the CEO or CIO.
  - Major market changes of direction, etc.
  - Whatever happens – it is unrealistic to not drop anything of a similar effort, otherwise we are just fooling ourselves.**

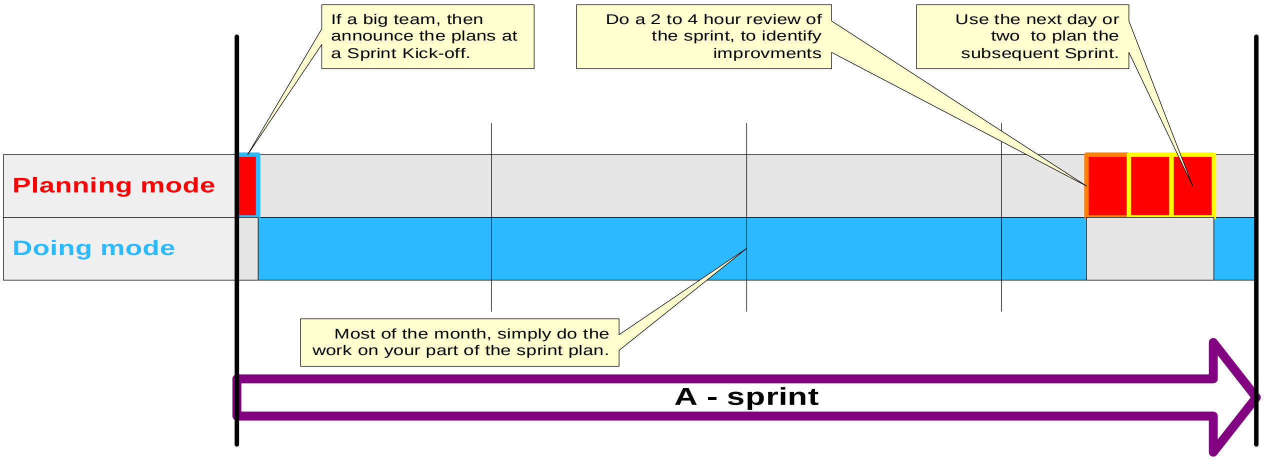
4. **Review at the end of the Sprint** – The only way you learn is from your mistakes and observations. No matter how under pressure the team is, it should allocate at least two to four hours a month to discuss and deeply understand how things did not work, how things could be improved next time. A constant feedback loop to constantly and iteratively improve. Also celebrate to successes. Don't only harp on the negative; value the positive.
5. **Plan for the next Sprint** at the end of this Sprint (its built into the previous plan). Dedicate at least a day. Get out of "Doing mode" into "Planning Mode."

** **This is where the concept of time-boxing comes in.** It is possible to take on more work (lets say 4 hours of work) by assigning one of the other tasks to time-box mode, where you do the best you can with the leftover time available to you.

Often 80% of a job takes 20% of the time, and is good enough to make a start. It is certainly preferable to make a start than to not doing anything at all, as it allows others to continue in the right direction, without holding them up and making them unproductive (which has an exponential unproductive effect.)

Sprint modes

Alternating between "Planning mode" and "Doing mode" is something people find difficult to do it appears (in general) – along the lines of getting out of "solution mode" into "requirements mode" – also difficult for people at times.


If a big team, then announce the plans at a Sprint Kick-off.
Do a 2 to 4 hour review of the sprint, to identify improvments
Use the next day or two to plan the subsequent Sprint.
Planning mode
Doing mode
Most of the month, simply do the work on your part of the sprint plan.
A - sprint

Planning modes in red have three main activities:

- To kick-off the Sprint and its objectives (mainly for a large team = > 15 people)
- To have a review session (heartbeat retrospective in Scrum terminology)
- To plan for the next sprint (planning session in scrum terminology)

Doing mode in blue

- To do all the work planned and assigned to each individual
- Also to ensure alignment of peoples time for collaborative workshops, etc.

References

[1] Wikipedia: http://en.wikipedia.org/wiki/Scrum_(management)