


Avoid technical debt and address software architecture in Scrum with the help of team architects

 ontheagilepath.net/2012/12/avoid-technical-debt-and-address-software-architecture-in-scrum-with-the-help-of-team-architects.html

By Sebastian Radics

Background



It took us some time to figure out a good way of handling architectural topics within our Scrum environment. In this post I'd like to share how we deal with technical debt vs. feature development, how teams contribute to the architecture and our way of syncing on architectural topics across the teams.

The current setup (brief description)

Each of the development teams works with a dedicated *team architect* – (s)he's a real developing team member. We conduct a weekly *Scrum of Architects* that lasts 1 hour to discuss architectural topics raised by the teams, architectural topics in general and to decide on so called *Tech stories*.

Some more details:

The role team architect

(S)he's a senior software developer having broad experience in software development and a deep knowledge of our current development stack. A profound interest in software architecture, good communication skills and a high interest in playing with new things is given too. In this role the team architect:

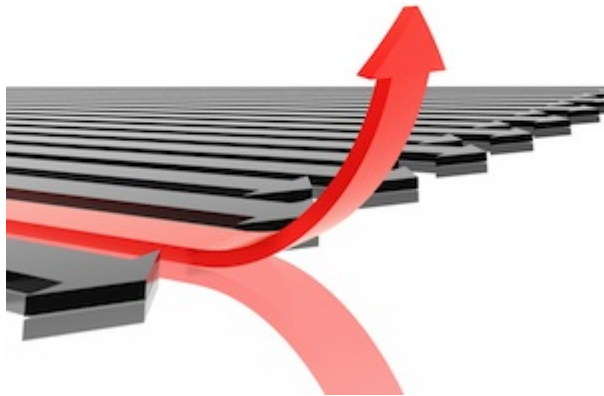


- guides and supports the team creating software designs and revising existing ones by suggesting possible refactoring or redesign necessities.
- enables decision making on architectural questions in the team and across teams
 - by sketching a design in the sprint planning and/or as sprint preparation + during the sprint
 - being involved in all architectural discussions
- plays an active role in code reviews
- fosters sync on architectural topics across teams
- creates and fosters approval for tech stories raised by the team
- supports the product owner in pre analysis of stories, defining business values and priorities for tech stories
- still helps developing important software parts (to stay in touch with the code and to see architectural areas)

To make it more transparent – the team decides on the solution and the team architect acts as an enabler towards finding the best solution. When an architectural decision needs to be made (s)he takes the responsibility.

What about tech stories?

It's a construct to distinguish business stories and stories related to technical improvements already by name. In the beginning we tried to “hide” efforts to refactor or redesign behind business stories – but it lead to much confusion for the product owner and later for the team too, as it was not really obvious why a story was estimated that high, especially compared to a similar story regarding functionality.



In addition we learned that the a product owner does not have the deep technical understanding to create these stories and make priority decisions on it.

Tech stories emerge out of the teams as they see a necessary refactoring, redesign or technical improvement (e.g. Java version upgrade). Often it's discovered during a sprint when taking an implementation step and discovering something new (especially in legacy areas without a deep up front analysis).

Making refactoring or redesign too fast and without questioning the benefit and best way of doing it and especially under the umbrella of a sprint commitment leads to a hurry. We avoid the hurry by taking a deep breath and creating a tech story for it.

Beside the description what the tech story is about the team architect has to add a business context (why is it worth doing this story) and a rough first estimation (is it a tiny, small, medium or large one). The business context helps to “compete” with the business stories and gives the product owner a better changes to put it a the right place in the product backlog.

Tech stories have an own lifecycle – that looks as following:

Tech story lifecycle

1. The team architect or the chief architect creates a tech story and adds it to the tech story backlog (an own backlog for all tech stories – see it as an incubator before a tech story is allowed to go into the product backlog for a team)
2. The tech story is checked by an approval team – if necessary changed and finally approved or rejected (based on discussion).
3. It's decided what team could implement the tech story (mostly tech stories come from a team – this way the assigning is obvious)
4. An approved story is added to the teams product backlog.
5. The product owner and team architect decide on the priority.

6. The tech story is estimated in the next backlog grooming session and the priority gets adjusted with the new information.
7. A tech stories goes in a sprint when it's the next story to pick up from the product backlog. The team architect helps the team to understand the background and moderates this part in sprint planning I+II
8. The implementation gets reviewed by the team architect and for bigger stories in the Scrum of Architects.

The Scrum of Architects meeting

Purpose of this weekly meeting is:

- to discuss architecture topics raised by the teams or raised from the chief architect or tech lead
 - it ensures a tech sync across the teams
 - common tech impediments get visible
 - it hinders having more than one team working on the same topic without knowing it
- have a tech story backlog grooming and approve/reject/request for change on not yet considered tech stories (the story creator explains the background in details – it's like selling the story)



In our context its scheduled weekly for one hour and is moderated by the chief architect. Attendees are – all team architects, our chief software architect and our CTO.

To ensure an good information flow regarding decisions and outcomes the meeting results are communicated by the team architects in their teams in time after the meeting (should (s)he forget it – the team will ask for the sync ;-)). In addition everyone can follow the wiki page with action points and the overview on topics discussed (by watching it). The tech story backlog is visible for everyone.

Everyone can follow and contribute

- The team has a strong voice through the team architect
- Architectural information is accessible

- via the tech stories and its store – the tech story backlog
- a tech blog
- public meeting minutes
- an active information sharing by the team architect
- The product owner can maintain the product backlog including tech stories with the help of the teams architect.

Conclusions



The current setup enables handling technical topics and managing (better avoiding) technical debt accumulation.

It's a transparent way – as it's visible by all involved parties.

Teams have an ensured architecture focus by having the team architect among them and can contribute and take responsibility for architecture.

Questions



Whats your experience with architecture in Scrum?

Do you use similar roles or a completely other way?

Can you help me spotting flaws in our way of doing it?