Combining Task Board and issue tracking software in agile development

Renaud FLORQUIN
FloConsult SPRL

Isabelle LECLERCQ
FloConsult SPRL

Agile development is sometimes difficult to introduce in teams. This article presents a way to help people new to agile finding their way in the iteration with a task-board combined with an issue tracking software.

Abstract

This present article is about introducing micro-planning practices to people new to agility, using a task-board practice, combined with an agile planning tool like XPlanner or an issue tracking software, like Bugzilla (Open Source) or Jira (Commercial).

We first present a first overview of the use of issue tracking software in iteration micro-planning management.

Then, we will speak of the advantages and problems using such tools in agile development, in particular when introducing such practices to people new to it, and how we managed combining it using tasks boards.

Finally, we will discuss of different implementations of the choices we present in this article, and of different things we should think of.

Introduction

Many of us become used (at least to) some agile practices in agile and iterative software development, like the scrum meetings, the pair programming, test-driven development,…

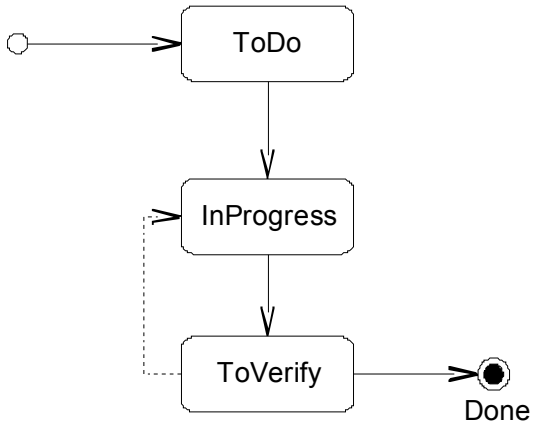
It is usually admitted that different planning levels coexist in software development, with at least two-levels:

1. the macro-one (gross-level granularity, in tasks in weeks or months, with the global milestones of a project; called “Release Plan” by Jim Highsmith in his “Agile Project Management” book [1])
2. the micro-one (at the iteration-level, with tasks in days or hours), made iteration per iteration. This is, still following Jim Highsmith [1], the “Next Iteration Plan”. This plan is of the responsibility of the team, and should be elaborated with all of its members, the tasks being “signed up for” by members, rather than “assigned to” by the project manager.

This “Next Iteration Planning” may be done using tools like XPlanner or Jira, but team members not used to agile or to collaborative team management may be lost when trying to get a view of the iteration planning. That is why we combined their use with the task-board practice (see [3]), as explained hereafter.

Using issue tracking software for micro-planning

Basically, tasks to achieve in agile software development follow the lifecycle:



Tools like XPlanner implement directly the follow-up of such development tasks, but issue tracking tools like Jira or Bugzilla may also be used for such a follow-up, keeping in mind that "issues" are not only bugs in software, but rather "tasks" to achieve.

Obviously, the challenge in such a use is to avoid an exponential creation of tasks, and also to manage tasks not directly related to coding.

We will come back later on to the first part of the challenge. To solve the second one, we may create "types" of issues, like for example:

- ✓ Management task
- ✓ Design/Architecture task
- ✓ Analysis/Test specification task
- ✓ User validation task
- ✓ Development (including unit test) task
- ✓ (Validation or Acceptance)Test task

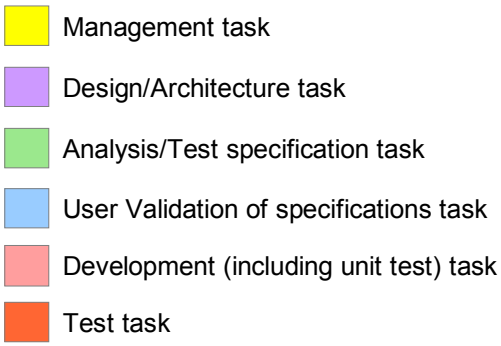
Generally speaking, each type of task may have a different type of assignee, and may possibly present variations on its lifecycle (more states, or more transitions,…), but keeping the same simple lifecycle is recommended at first.

Task-board; task types and lifecycle

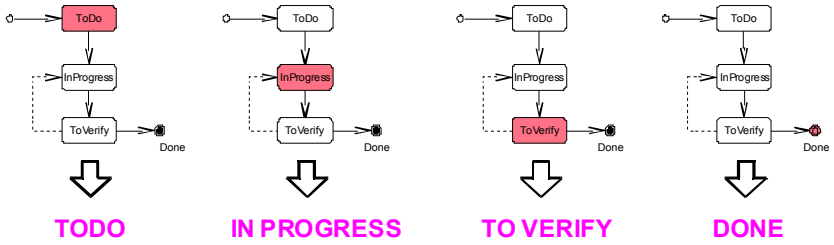
As said in the introduction of this article, people new to agile development are sometimes a bit "lost" at the beginning and assigning them tasks through software does not help them to find their way in the development. Obviously, the daily scrum meeting helps, but we found very useful to base this scrum meeting on a "concrete" support, that is, the task board (see also [3]), with post-it used to represent the tasks.

We mirror on a white-board (or on a wall) the identified tasks to achieve for the *current* iteration, with different colours to distinguish the type of tasks, and different columns to show their current state of resolution.

For example, coming back to our issue “types”, we may have chosen the following “legend” for our task board:

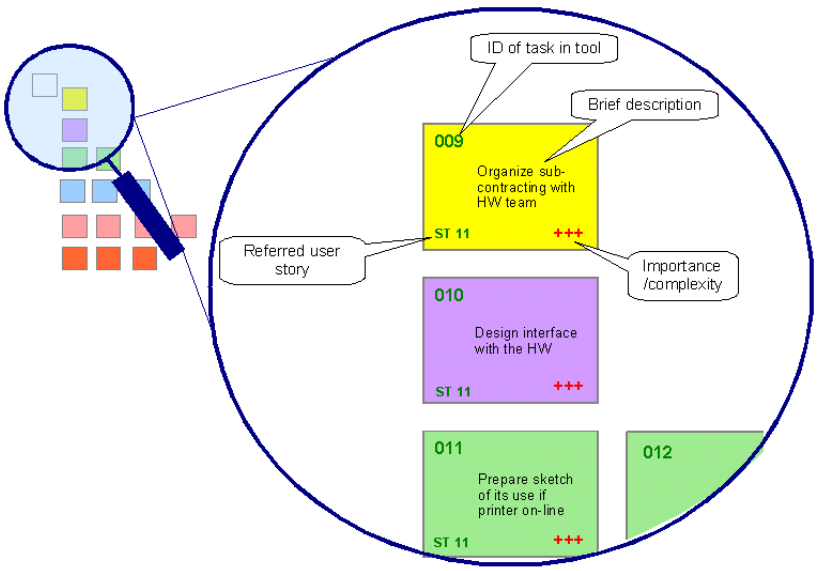


And regarding the columns, we have the correspondence to the tasks lifecycle:



What to set on a task paper representation?

Our goal was obviously to avoid too much redundancy between the paper representation of tasks, and their software image, while offering a concrete view of tasks. Thus, we limited ourselves at writing only the id of the task in the tool, a summary sentence, and optionally the complexity and/or the user story reference:



…

The complete definition of the tasks, the dependencies between them, the dates,… are to be seen in the issue tracking tool, while the task board is a visualization of who is in charge of what at the current time of the iteration.

Task board; "IN PROGRESS" columns per team member

The "IN PROGRESS" column contains sub-columns with team members, client (here = "CLI"), and other impacted people (here = "OTH"), in charge of tasks.

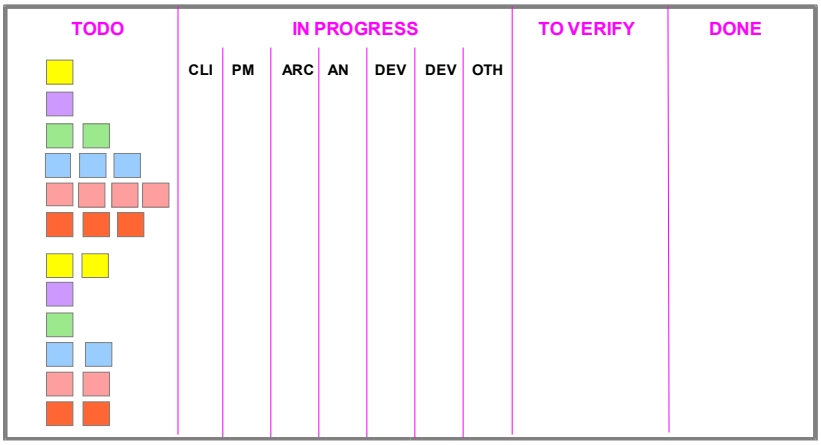
During the daily scrum meeting, the team members "apply for" tasks, obviously taking the tasks dependencies into account, and possibly engage themselves on the work duration. This is visually represented by setting the corresponding post-it in the people's column "IN PROGRESS".

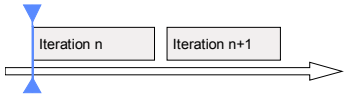
There is generally speaking a link between the team member role (e.g. a DEV = developer generally is in charge of development tasks), but this is absolutely not mandatory, as the tasks are collectively and on a voluntary basis assigned.

To avoid too much complexity of this task-board, we found that the only place where this differentiation per team member is really useful is the "IN PROGRESS" one, as the "TO VERIFY" is generally done by a more limited group of people.

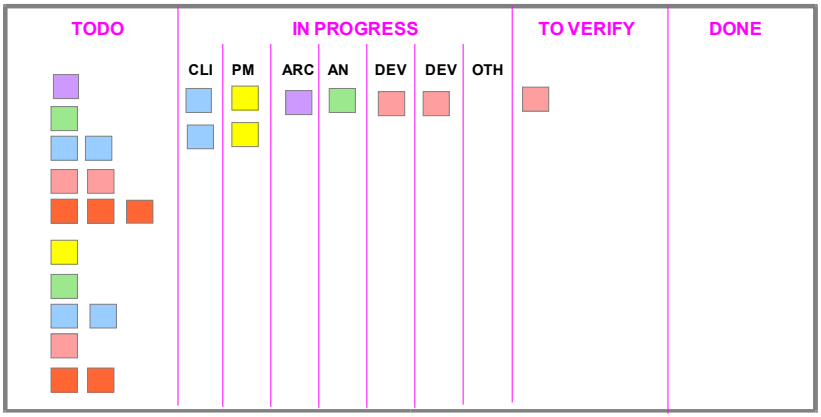
Typical task-board views at different moments in the iteration

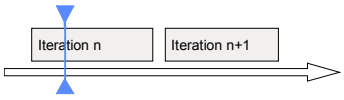
At the iteration begin, we have the identified tasks in the "TODO" column:

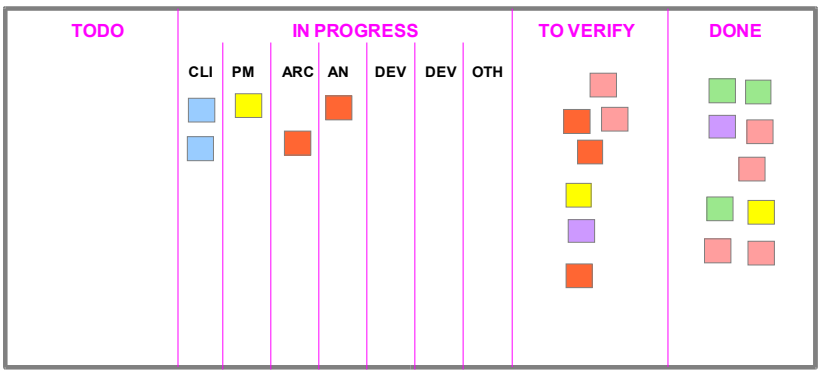


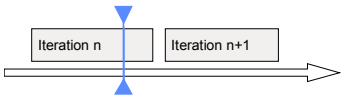


Then, it evolves:

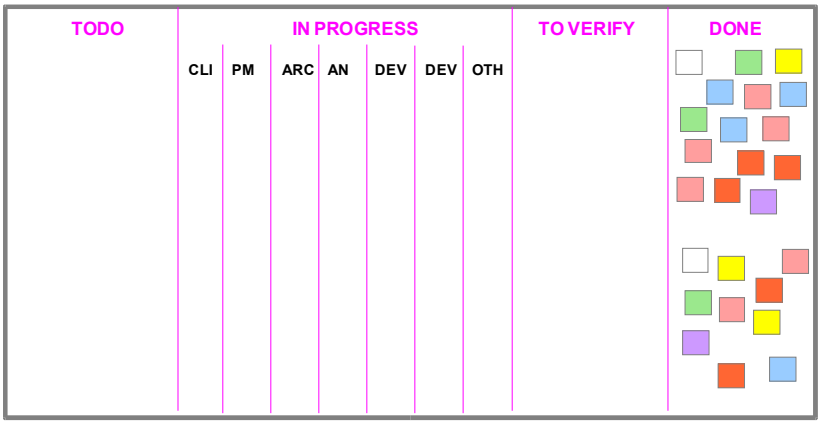


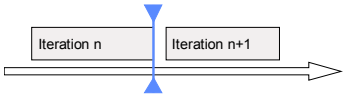




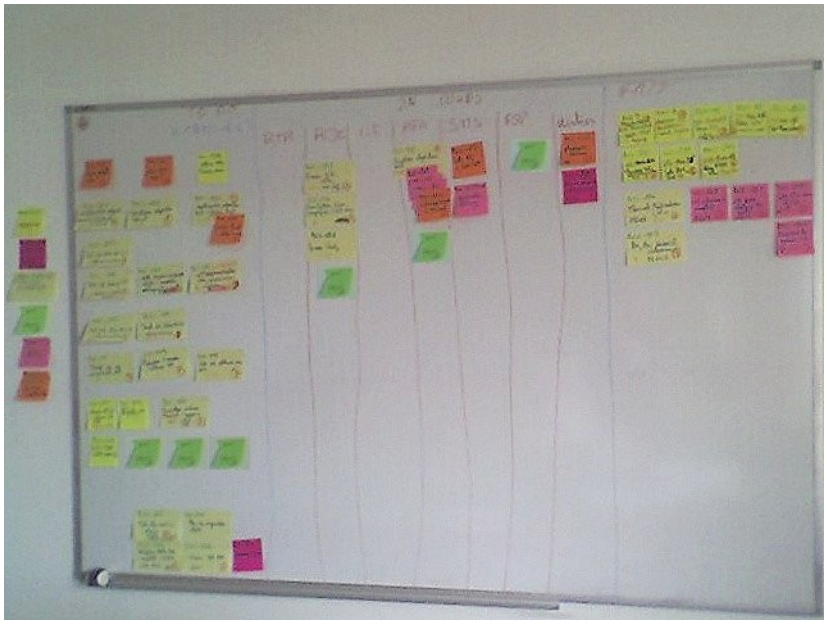


The goal being to have at the end of the iteration all the tasks done:





Photography of a task board during an iteration



Pros & Cons of this technique /recommendations

1) Do not put all the tasks on post-its

Putting all the tasks entered in the tool (may be 150 for one iteration for a 4-members team) is not practical, and also not interesting if we want to keep the task board simple. We put only the tasks that represented a significant amount of work (minimum some hours), or sometimes we put more than one issue one the same post-it (typically in the bug correction stage)

2) We do not pretend that there are no task creation/move/deletion during an iteration

We presented a very static view of the task-board, with the issues created at the beginning of the iteration, evolving and finally closed. This obviously does not imply that we did not created issues during the iteration, or moved some to the next one, or suppress some, as in any software development. These modifications, if done taking into account the agile principles, are simply reflected on the task-board and logged in the tracking tool.

3) Choose knowingly the task types for the product backlog

Having other types of tasks that development ones may imply different sorts of tasks backlog (e.g. the analysis or test tasks typically stay longer open than development one, which themselves stay longer open than the bugs). We were used to produce the statistics with all the issues types, but this implied a product backlog that seemed to evolve less agilely than the usual one.

Conclusion

We found this combination of task tracking software/task-board visualisation very useful to get people new to agile involved actively in the scrum meetings, as it offered a day to day concrete view of who was doing what.

Some agile planning tools or even issue tracking software offer such visualisation possibilities on the computer screen, but we found that people still prefer and better understand the organisation with a (even very simplified) concrete view such as the task-board one.

On middle-term, the challenge of the scrum master and of the team is to keep the task-board alive, usable and pragmatic, for example keeping in mind the recommendations done.

Acknowledgments

Many thanks to our friends Antoine CHOPPIN and François ROUCOUX for their valuable comments.

References

[1] Agile Project Management, by Jim Highsmith, Addison-Wesley, 0-321-21977-5

[2] Scaling Software agility, by Dean Leffingwell, Addison-Wesley, 0-321-45819)2

[3] “Task Boards”, Mountain Goat Software, http://www.mountaingoatsoftware.com/task_boards