

Agile Management with uScrum

Joaquim Baptista — Altitude Software — Portugal — px@acm.org



Altitude Software develops a software suite for contact centers: "Screen pop" for contact center agents handling calls, email, chat; Automatic dialling, self-service for calls.

Technology:

Client/server emphasizing Windows, supports AIX and Linux. Integrates with major telephony switches (PBXs), major software suites (such as SAP, Siebel).

Supports customer developments in proprietary programming language, C, C#, Java, others.

If estimating a hard task takes almost as long as doing it... How do you manage uncertainty?

uScrum (uncertainty Scrum) is an agile process developed by a small team at Altitude Software to manage the process of writing user documentation. uScrum manages uncertainty and the unknown, allowing writers to quickly react to changing conditions.

uScrum uses orders of ignorance to understand the difficulty of tasks, allowing the team to effectively prioritize regular work together with difficult creative work.

Download full paper at <http://pxquim.com/en/>

How hard are tasks?

Classification inspired by Philip Armour's work: 0oi means 0th order of ignorance.

- 0oi** Writers know everything needed to complete the task.
- 1oi** Writers know what they need to write, but the information must still be gathered from developers.
- 2oi** Writers know the problem to solve, but the team has never solved a similar problem.
- 3oi** Writers try to solve a hard problem but are not sure how to proceed.

Wiki artifacts

Backlog (ToDo) pages

Wiki pages hold task entries ranging from simple bug fixes to large or vague tasks. Sample task:

5-10d 2oi skill.sigdoc Write a paper for SIGDOC'08 that explains the management practices of the team.

Sprint (Do) pages

Wiki pages record decisions and events of a sprint.

Kick-off: Period of the sprint. Rationale behind choice of tasks for the sprint. Tasks assigned to each writer.

Real work: Notes taken by writers during the sprint. Unplanned work.

Last week: What to publish, what to present. Team achievements. Compare estimates with actual efforts, with notes on outcomes.

Task scheduling for sprints

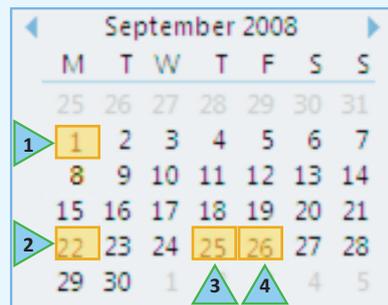
Different mixes of tasks originate different kinds of sprint, depending on what tasks fill the sprint *bucket* of effort. Writers are over-committed and micro-manage their own work. Team holds stand-up or ad-hoc meetings as needed.

Get things done:
Writer has tasks at 0oi and 1oi. Many tasks finished at the end of the sprint.

Try with fallback:
Writer has 2oi task complemented with less important tasks at 0oi and 1oi. Writer advances the hard problem as possible, fills slack with predictable work.

Just try:
Writer (or writers) have a single 2oi task. Writer advances as much as possible during the sprint.

Cannot go back:
Several writers work in a disruptive problem that makes the documentation set unpublishable.



- 1** **Sprint kick-off meeting**
(First week, Monday) Assign tasks to writers, note rationale.
- 2** **What to publish meeting**
(Last week, Monday) Negotiate what to publish, How to handle unfinished work, What to present.
- 3** **Sprint review meeting**
(Last week, Thursday) Writers present significant work to each other.
- 4** **Sprint retrospective meeting**
(Last week, Friday) Review metrics of the sprint. Note team achievements. Discuss issues, Propose improvements.

Story of uScrum

uScrum has evolved and been applied since 2004 by a team of 3 to 5 writers following two practices.

Practice: Metrics spreadsheet

Enabled team to learn where does the time go.

Page for each month, rows for days, columns for tasks, minimum trackable time is 0:30.

Practice: Yearly team improvement meetings

Enabled team to evolve uScrum in yearly steps.

Yearly meeting with 4 questions answered privately and openly discussed:

- What keeps you from being as effective as you would like to be in your position?
- What keeps the unit from functioning as an effective team?
- What do you like about the unit that you want to maintain?
- What suggestions do you have for improving the quality of our working relationships and the functioning of our unit?

Scrum assumptions

uScrum assumptions

	Scrum assumptions	uScrum assumptions
Product backlog	The top of the product backlog is fully prioritized and estimated.	2oi tasks cannot be reasonably estimated, although the team can estimate the effort that it is willing to invest at solving a task in each sprint.
Sprint goal	Team demonstrates an objective to the product owner in the sprint review, based on a realistic sprint backlog.	Writers show results for the effort invested. Overbooked writers select what work gets done during the sprint.
Team	Work must be coordinated among team members.	Work is mostly independent of other writers.
Sprint tasks	The sprint backlog can be broken down into 4 to 16 hour tasks in the first day of the sprint.	Finding out what to do or how to do it is a large part of the work that cannot be fully anticipated.
Sprint progress	A burndown chart can track the progress through the sprint.	There is nothing interesting to count down during a sprint.
Obstacles	The team can complete its tasks because it has the necessary skills and resources from the beginning. The Scrum Master can quickly remove any obstacle.	Some obstacles simply cannot be removed or anticipated by the team, such as a developer going on a two week vacation immediately after finishing a project.