

# AGILENOMICS™ RAPID TRAINING SERIES

## **TRAINING PUBLICATION #1: ACCIDENTAL DISCOVERY - IS IT POSSIBLE TO ACCELERATE THE STABILIZATION OF A BRAND-NEW SCRUM TEAM'S VELOCITY BY DELAYING THE INTRODUCTION OF STORY POINT ESTIMATION?**

**PROBLEM STATEMENT:** A new scrum team, especially consisting of novice developers, can take several sprints to stabilize their story point velocity. The following technique has proven useful in reducing this timeline, by separating the earliest part of the learning cycle from story point confusion.

**BACKGROUND ASSUMPTIONS:** Most agilists will likely agree on the following generalizations as being almost factual:

- 1) A brand-new team embarking on a brand new project needs time to learn before they can reach a sustainable pace.
- 2) There is general confusion, especially when novice / junior developers are involved, as to what a story point even entails, let alone how it relates to the "uncharted waters" of the first sprint.
- 3) It is not uncommon for it to take several sprints for a brand-new team to have its first sprint in which all work is completed and accepted.
- 4) It is not uncommon for new teams to take a LOT longer to understand how to reduce their batch sizes in order to maximize throughput.

It takes "doing" the work to "understand" how to estimate future work. This is not rocket science, but for some reason, methodologists have insisted for years on prescribing "story point" use from day 1. At a certain client engagement, an Agilenomics™ consultant experimented with a far different approach, one that may shock the Agile purists, but the results were so striking that we felt they needed to be published.

### **STEP 1: DECOUPLE "INITIAL LEARNING" FROM "STORY POINTING"**

Traditionally, for those who still believe in a "sprint 0" as a period of time to train and prepare a new scrum team to a state where they can begin sprinting normally, the following high level "exit criteria" would apply:

- a) Ensure the team has sufficient backlog to start sprinting
- b) Ensure the team has an environment and tools that enables them to deliver value
- c) Ensure the team has the skills they need to do the work (perhaps some training is required in sprint 0)
- d) Ensure the team has been trained in Scrum Methodology

- a) Ensure the team understands how to estimate future work (i.e. story point estimation)

And this is precisely the problem... “e”...time to ask one of the infamous “5 why’s”...Why would we do this if we already established further above that there is confusion on day 1 on what a story point may entail? Why not try a bold experiment that hits two birds with one stone...

**REVISED STEP 1) LET THE TEAM LEARN THE REALITY OF THEIR ACTUAL WORK ENVIRONMENT FIRST, AND WE CAN WORRY ABOUT TEACHING THEM STORY POINTS A BIT LATER.**

- a) Ensure the team has sufficient backlog to start sprinting
- b) Ensure the team has an environment and tools that enables them to deliver value
- c) Ensure the team has the skills they need to do the work (perhaps some training is required in sprint 0)
- d) Ensure the team has been trained in Scrum Methodology
- e) Pick a small story, and conduct a mini-sprint of 2 days (See publication on “how to conduct a mini sprint”)
- f) Based on the Retrospective, repeat step “e” until the team successfully understands how to size stories small enough to be completed in a 2 day sprint.
- g) Upon a single successful occurrence of “f”, the team will experiment with a 5 day sprint.
- h) Only repeat a 5-day sprint if the team fails to complete more than 30% of the stories selected.
- i) As soon as a 5-day sprint is completed with greater than 70% story acceptance rate, the team is ready to exit “sprint 0” mode, and move to normal 2-week sprints.

Note: Up until this moment, the team will be working on user stories that are in normal user story format, have normal acceptance criteria, are tasked in hours, but that do NOT have any story points assigned. (Some purists may disagree already, but please keep on reading)

**STEP 2) INTRODUCE STORY POINTING AS AN EXERCISE THAT CORRELATES ABSTRACT ESTIMATION TECHNIQUES WITH KNOWN, COMPLETED WORK**

This is another controversial moment. After the team has actually completed several stories of known work, now is the time to “story point” the COMPLETED work. Yes, we are going to resurrect stories that are marked “DONE” for the purpose of accelerating the team’s capability to estimate and reach stable velocity.

The “reverse-estimation” Process is simple:

- 1) Call the team to an estimation session
- 2) Present the Completed story cards on the wall / board / screen

- 3) Ask the team to agree on what the Least Complex story they completed was
- 4) Ask the team to agree on what the Most Complex story they completed was
- 5) Ask the team if they feel there is a theoretical category of stories that could even be Simpler than the one they have marked "Least Complex" today....if the answer is yes, have them call that a "1", and estimate the "Least Complex" one against that. Otherwise, the "Least Complex" story becomes a "1"
- 6) Ask the team to pick a complexity that they all agree could not possibly fit into a 2 week sprint (13, 21, etc depending on the team's choice). Make a team agreement that they will never bring such a story into a sprint without breaking it down.
- 7) Ask the team to estimate the Most Complex story they completed against the "1" and the "theoretical maximum" for a 2 week sprint.
- 8) Ask the team to estimate the remaining completed stories against the other two they just "pointed" using the modified Fibonacci sequence we are all familiar with in Agile.

### **STEP 3) NOW, THE TEAM IS READY TO STORY POINT THE REMAINDER OF THE BACKLOG**

Have the team start estimating the prioritized backlog at this point, because they can do relative estimation much quicker and more accurately now that they have "learned" from the work that they already completed in the mini-sprints.

It is helpful to keep a visible chart of the points as well as those "done" stories up in the team room during the first few estimation sessions for reference purposes, until it becomes second nature.

### **STEP 4) EXECUTE SPRINT 1 & ITERATE FOR STABLE VELOCITY**

This experiment was run twice, with 2 different teams.

The first team became the highest performing agile team in the company within 90 days.

The second team was actually an existing team, but they transitioned to a brand-new project with so many unknowns, the team had to revert to a "sprint 0" state, and we made use of the mini-sprint and delayed story point techniques as above.

We found that both teams ended up with significantly less variation over the first 3 "real" sprints than what we typically experience with new teams, and they actually reached a stable velocity by "real" sprint 3, whereas other similar teams we have observed in the past might take an average 5 to 6 sprints to reach this point. This does not mean their velocity did not continue to increase over time, but they were making their commitments consistently much earlier than other teams we had observed in the past. In a world where MVP learning cycles are increasingly important to make longer term investment decisions, we feel that this experiment is worth further examination by our colleagues and experts in the market.

## EXAMPLES OF FAST LEARNING CYCLE:

Here is a summary of how many “Learning Cycles” occurred with this technique:

- In the case of the brand-new team above, on day 2 of the mini-sprint, management decided on replacement of 4 resources. This was an unintended consequence, but the mini-sprint made it transparent that these team members presented by a particular 3d party vendor did not possess the skills that were expected and outlined in the contract. To mitigate long-term risk, management was forced to act once the problem was identified in practice. The “mini-sprint” acted as a prototype MVP for exposing a contractual and quality issue that would not have surfaced for much longer under normal circumstances. The new team members arrived the next week, and despite the initial shock, the disruption was actually minimized, and the longer-term investment was protected.
- Both teams focused on learning first, and were able to estimate in story points later, in a more meaningful and effective manner. We did not lose the story point “burn-up” data by not having those points available initially, because all 10 or 12 of those completed stories were pointed within 30 minutes by the team after the fact anyway, **as part of a learning exercise** for them to understand how to point future work more accurately.
- We were pleasantly surprised to see that the stability of a team’s velocity appeared to accelerate as a result of this technique, and despite the appearance that the techniques may be introducing anti-patterns, if these results can be repeated and verified in the field, the underlying economic benefits cannot be ignored.