Promoting Teamwork for Devs in a SCRUM Environment

Author: Justin Pope

# Preface

Our current SCRUM team operates on two-week sprints with normal ceremonies. Our current team is the following:

***Product Owner -*** Robert Hobbs

***Developers -*** Reid Wilson

Daniel Arwe

Justin Pope

***QA –*** Jennifer Hickey

Walter Martinez

***SCRUM Master –*** Joe Ebeling

***Designer -*** Charlie Bradley

Our normal cycle of work in a sprint contains with work being committed to the sprint during Sprint Planning. During this commitment, developers are assigned individual stories to work on and the individual plans how the definition of done will be completed. The developer is then responsible for the following events:

**Pre-Story Meeting** – Not always necessary, but a conversation with QA, Product owner and other necessary members on how the story will be completed and to ensure acceptance criteria is meet as well as testing concerns are voiced.

**Development** – This is where the bulk of the individual developer is accomplishing the main bulk of development on the software product.

**PR Review** – This is where the individual developer creates a pull request into the main branch of the repository. Other developers are responsible for critiques on the suggested code changes and other comments.

**Testing** – The PR is completed, and our Azure Pipelines create builds and releases for our environments. Then QA runs tests within the QA environment to confirm acceptance criteria is met. If QA finds issues, they will mark their findings and pass the story back to development for bugs and kinks to be squashed or ironed out. If QA finds no issues, then the story is passed to Product Owner to review.

**PO Review** – The Product Owner reviews the work that was completed and evaluates if more is needed for the product. Other edge cases could come up in this step of the process or the new ‘feature’ could be deemed as not ready to push, but for the most part this is just a litmus test.

Overall, our team works with good efficiency on completing work that is assigned to us in each sprint.

# Problem Statement

“There is a lack of teamwork amongst developers when it comes to complete committed work.” -*Justin Pope*

Since the stories are assigned to individuals, the following tendencies are created:

1. **Siloed Development** – Developers primarily focus on their assigned work, reducing team collaboration and collective ownership. This is also seen as that if a story is looking like it is going to carry, developers do not all try and help get the story over the finish line. All committed stories are all developers’ responsibility.
2. **Knowledge Gaps** – Developers have limited insight into stories that are worked on by others. This creates barriers that hampers the team collectively.
3. **Missed Opportunities** – The team lacks a shared approach to improving the product collectively, limiting innovation and cross-pollination of ideas.

# Proposed Solution

To promote collective ownership and teamwork amongst developers, we should shift from individual-assignment model to a team-collaboration model.

## Implementation

Proposed methodologies:

1. **Story Swarming:**Stories will be assigned to the developers collectively and together will be tasked out. Designs should be collectively built and democratically decided on the best approach. Tasks should be broken down to tasks that would achieve the smallest unit of work that is needed to carry out the shared design. QA and designers are intended to collaborate in tasking to complete each story.
2. **Cross-functional Pairing**  
   Pair programming or short burst of individual coding should be used to complete the intended solution. QA should be brought in and participate in development to run through acceptance criteria early in the process. Designers can be brought in when necessary.
3. **Pull Requests**Since there is more collaboration to development, pull requests should be relatively quicker. Responsibility for review should be given to the developer that did the least amount of development or each developer on the team should approve the Pull Request.
4. **Testing**Since QA was brought into development, testing should theoretically be quicker and decrease the amount of back and forth between QA and Developers.

## Intended Results

Implementing the plan above would hopefully result in the following:

1. **Increased Collaboration** – Developers should work together to come up with solutions that would achieve a stories acceptance criterion. This would remove knowledge gaps and lower the chance of missing opportunities.
2. **Improved Product Quality** – When developers all have a vested interest in each story, there are more chances for easier feedback loops which would increase the quality of the product.
3. **Higher Engagement** – Each developer will have a higher level of engagement which in return will increase buy in and decrease knowledge gaps.

## Possible Complications

The intention of this proposed idea is to create more teamwork across our SCRUM team. The following counter intentions are things that we want to be concerned about:

1. **Alpha Character**   
   Every voice should matter in decision making. One person should not be calling the shots, and democratic decisions should be made on design. When there is a dominant voice, assigning individuals responsibilities for task to work on alone and then present to the team.
2. **Team Mold**   
   We do not want to just collectively have our selected areas of development that each individual works on. Everyone needs to be versed in every aspect of our application. When there is an aspect of something that someone else is ‘weaker’ in time should be invested in raising that individual up. We are only as strong as our weakest link.
3. **Reduced Productivity**   
   There will be time that will be needed to implement this strategy and iron out good processes for this to take hold. At each sprint retrospective, it is important to identify pain points to this process and be able to refine them. The following adoption plan will implement a slow plunge so that issues are encountered and resolved quickly and efficiently.

# Adoption Plan

The following suggested timeline should be taken to adopt this plan.

## Phase 1: Partial Adoption

Start: beginning of Sprint on January 15th

Duration: 2 sprints

Fifty percent of stories should implement this collaboration model.

## Phase 2: Full Adoption

After the completion of phase 1, full adoption should be taken.

## Retrospective

Sprint retrospectives are critical to identify pit falls into this plan to be able to course correct. Failure to do so will result in failure in implementation.