# Table of Contents

# How Do We Ensure Critical Work Does Not Get Dropped as We Kick Off Teams?

## Premise

One of the fears that many organizations have is that, as they “go Agile” and form Teams, important things will get dropped and things will get a lot worse for our customers before they start getting better. This is a well-founded business concern and needs to be addressed. The problem is that many organizations worry about “pulling the trigger” to do the agile transformation until they know how all the existing work is handled. This is a problem because:

- In many instances the people “designing” the Teams do not have a complete grasp of how work actually is done so they are really unable to come up with a solution that will work. For example, often the some critical thing gets sorted is not via the documented procedure, but rather through personal contacts once an issue has been raised. As we structure Teams as part of the agile transformation and we try to build a new procedure / approach to deal with these situations, then we could end up not only designing a system that does not work, but also increasing the chance that we break what has worked in the past.
- In many instances the people “designing” the Teams are new to Agile, and so don't have a complete grasp of how work flows in the new system, have self-organization works (in a practical vs theoretical sense) which means that they will make assumptions about the new process based on the old way of working. For example, if there was a supervisor assigning work for critical items in the past to ensure “nothing is dropped”, the assumption is that we need something like this in the future as well. The result is that you end up unintentionally reinforcing old ways of working just as you are trying to change how things get done.
- In many instances creating new procedures, handling the management of change required, working though the decisions and so on takes a long time. This means that while we are creating systems that might (probably) won't work, we are going to delay the formation of Teams (“going agile”). Assuming we are doing this transformation for good business reasons (eg improve customers happiness, faster time to market, improved quality / reliability, improved quality of life, etc) then this delay materially effects the outcome expected.

OK, so interesting, but we still have the business concern - how do we ensure that critical work is not dropped?

## Guidelines

When we form a Team one of the guideline that is suggested is that Team members “take their work with them” (see How Does a Team Initially Get Control of Work? for other guidelines). In other words, if you were the “on-call” person for this set of technologies in the past (and in the absence of any other change) you are still that person when you get on to the Agile Team.

Why do we do this? One reason is to ensure that we are increasing overall understanding of all the work

that is happening in the team. But other reason is to ensure that we don't lose work.

As we form Teams we start to track work explicitly that the Team is working. We also typically also define a “scope” for that Team. So what happens if the work a Team member brings with them is not part of the scope of the Team. A couple of things could have happened:

- Perhaps the scope of the Team is wrong and we need to adjust that.
- Perhaps the scope is correct, but the work needs to be transferred to a more appropriate place.
- Perhaps the work, given current prioritizes, should not be done right now, and so should be scheduled at a later date.
- Perhaps the work actually is not that important and we should not be doing it all.

Irrespective of what the outcome is, we need to ensure we know what we do with the work and we need to be explicit and visible in the outcome. In particular we need to insure that if the work is to be done by someone else, that we have a warm hand-off to ensure that this has actually happened.

The general approach determine disposition of the work via a set of filters:

1. Should this work be done at all? If no, then inform however asked for it, and move on.
2. Is this work within the current scope of the Team?
    1. If yes, then work through standard Team prioritization (typically through the Product Owner for the Team). Note that like all prioritization, the resultant scheduling requires communication with the relevant stakeholder. For example, if the stakeholder is expecting something immediately, but priority of this item is low in relationship to other things the Team is working on, then the stakeholder has to be informed.
    2. If no, then determine who the work belongs to and have an explicit hand-over of not just this work, but also future work of this type. It will then be up to that Team to prioritize and schedule the work and work with stakeholders.

# Benefits of This Approach

There are a number of benefits to using this approach to determine where work goes to:

- It is an incremental approach requiring no “big bang” upfront design of new processes (and so is actually an agile approach to the problem:-))
- It respects existing ways “really get done around here” ensuring that there is a warm hand-off in the event of change in process.
- It begins the process of letting the people who do the work have the most say in how the work is done starting that very agile approach of “decentralized decision making” building toward a “self managing” Teams.
- It can be done immediately, whether you are working a single Team transformation or a larger multi-Team and / or organization transformation since there is no need to wait for the design of new processes.

# Want to Know More?

- How Does a Team Initially Get Control of Work?
- This is a use (modification?) of what Henrik Kniberg calls the "Bun Protocol"

FAQ, FirstSprint, WIP, dropped, Team