Added Complications

One of the great things about Scrum is not that it's perfect for solving every problem but it is pretty good ant shining a light on the problems you didn't know you had. The Stacey diagram, originating in the nineteen nineties describes differing problem regimes, the same regimes used on the more familiar Cynefin framework.

Scrum is a framework with feedback mechanisms built into it and hence is designed for solving problems in the Complex domain. This is the domain where you are not sure of your solution, but you keep attempting to provide answers until inspection confirms that you are correct.

This is fine for a tech start up but many of us 'live' in a work world where the products have been around for a while and have a lot of moving parts, both in a business and technical sense. This is Stacey's complicated regime, where you have to pull things apart to understand how to change them or have to have deep knowledge of the system to do so.

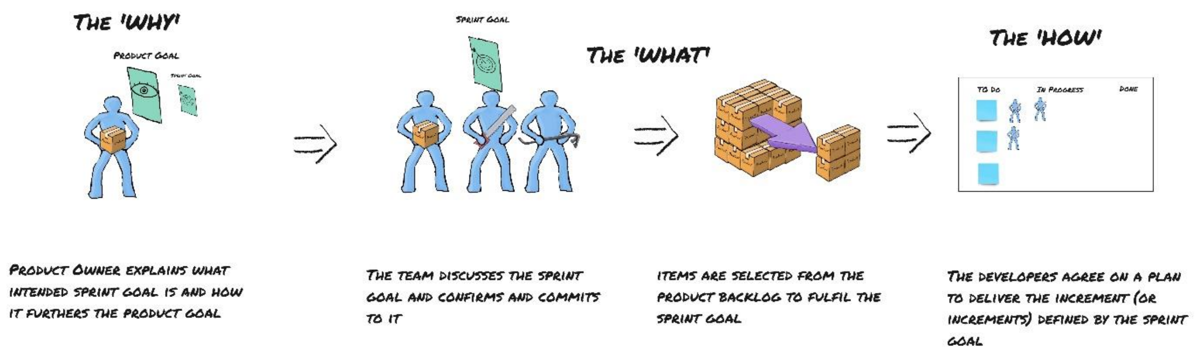
I've written before about the phenomenological effects of utilising Scrum in the Complicated world. Certainly, in the context of software development, it is easy to accumulate technical debt. As functionality is used by differing stakeholders then usage and 'requirement' defects will be found, or new enhancements requested. Within the Stacey model there is a regime deemed Chaotic where it is up to the problem solver to react to the current situation with as much immediacy as possible to reach an answer. Again, when a stakeholder pressure tests a product in some way, further 'defects' will be reported that may have a time urgency for said stakeholder. All of which somehow needs to be poured into a Sprint if Scrum is being used.

It is easy to say as a Scrum Master that all of these should be in a single Product Backlog.

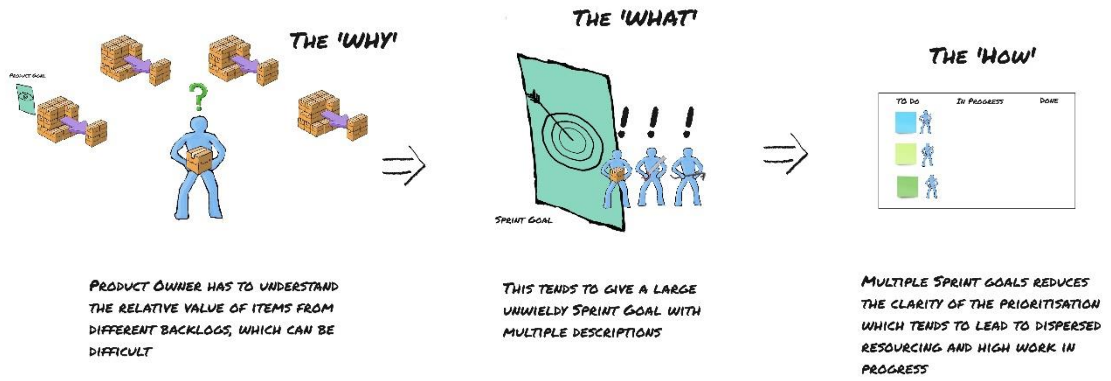
So, I'll say it. They should.

Choosing to adopt multiple backlogs to rationalise this problem can make it confusing when Product Owners need to prioritise one 'flavour of item' against another. The qualitative effect of this can often be seen in Sprint Planning when trying to construct a Sprint Goal or even order the work within a Sprint. This is visible when Sprint goals are just a list of Product Backlog items of differing types. This can lead to high Work In Progress as the Sprint just becomes a capacity exercise instead of one of value. But if you are stuck in this type of world is there anyway to mitigate such a problem?

It is clear within the Scrum Guide how Sprint Planning should be done. It consists of three parts. The Product Owner explains why the next piece of work will get us closer to the Product Goal, the team will then decide what has to be taken from the Product backlog to achieve this and then the developers work out a plan to execute the work.


THE 'WHY'
PRODUCT GOAL
SPRINT GOAL
THE 'WHAT'
THE 'HOW'
TO DO
IN PROGRESS
DONE
PRODUCT OWNER EXPLAINS WHAT INTENDED SPRINT GOAL IS AND HOW IT FURTHERS THE PRODUCT GOAL
THE TEAM DISCUSSES THE SPRINT GOAL AND CONFIRMS AND COMMITS TO IT
ITEMS ARE SELECTED FROM THE PRODUCT BACKLOG TO FULFIL THE SPRINT GOAL
THE DEVELOPERS AGREE ON A PLAN TO DELIVER THE INCREMENT (OR INCREMENTS) DEFINED BY THE SPRINT GOAL

In the complicated regime however, things can get a little confusing. With multiple backlogs the ordering becomes difficult, the Sprint Goal gets diluted containing multiple items which often leads to a plan with resources dispersed across the board

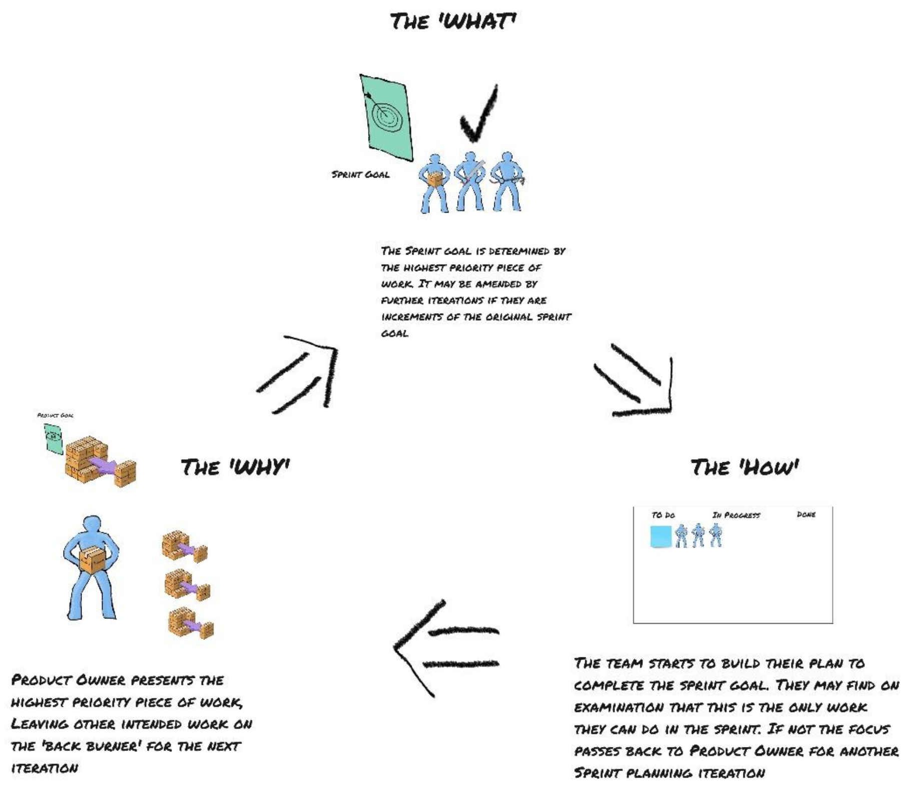

PRODUCT GOAL
THE 'WHY'
?
THE 'WHAT'
! ! !
SPRINT GOAL
THE 'HOW'
TO DO
IN PROGRESS
DONE
PRODUCT OWNER HAS TO UNDERSTAND THE RELATIVE VALUE OF ITEMS FROM DIFFERENT BACKLOGS, WHICH CAN BE DIFFICULT
THIS TENDS TO GIVE A LARGE UNWIELDY SPRINT GOAL WITH MULTIPLE DESCRIPTIONS
MULTIPLE SPRINT GOALS REDUCES THE CLARITY OF THE PRIORITISATION WHICH TENDS TO LEAD TO DISPERSED RESOURCING AND HIGH WORK IN PROGRESS

Scrumfall, the phenomenon of teams finishing a majority of the work at the end of the sprint, is one of the banes of a Scrum Master’s life and inevitably the cause of high work in progress. But this tends to be the last domino that falls, coping with the problem of multiple backlog thinking.

So back to our original question, is there any way to mitigate this problem by changing Sprint Planning? The Scrum guide says no, because the Scrum guide says that the only way to do Scrum is to follow the instructions written therein. But since you may already have pulled the pin out of the grenade there’s not a lot of sense in keeping hold of it to spite yourself.

We may have to break the law a little bit, but we will do it by reinforcing one of the foundations of Scrum. The idea of iterative and incremental delivery.

Instead of having the linear ‘Why’, ‘What’ and ‘How’ segments executed in order, we’ll take a vertical slice through it and iterate for each backlog.


THE 'WHAT'
SPRINT GOAL
THE SPRINT GOAL IS DETERMINED BY THE HIGHEST PRIORITY PIECE OF WORK. IT MAY BE AMENDED BY FURTHER ITERATIONS IF THEY ARE INCREMENTS OF THE ORIGINAL SPRINT GOAL
PRODUCT GOAL
THE 'WHY'
THE 'HOW'
TO DO
IN PROGRESS
DONE
PRODUCT OWNER PRESENTS THE HIGHEST PRIORITY PIECE OF WORK, LEAVING OTHER INTENDED WORK ON THE 'BACK BURNER' FOR THE NEXT ITERATION
THE TEAM STARTS TO BUILD THEIR PLAN TO COMPLETE THE SPRINT GOAL. THEY MAY FIND ON EXAMINATION THAT THIS IS THE ONLY WORK THEY CAN DO IN THE SPRINT. IF NOT THE FOCUS PASSES BACK TO PRODUCT OWNER FOR ANOTHER SPRINT PLANNING ITERATION

Starting with the highest priority backlog (I'm going to assume it's the Product Backlog!) the Product Owner will explain why the projected work is high in value and the team then formulate a clear Sprint Goal based on this. They will then formulate a plan for how to build increments for this goal. If the developers then judge that they have enough 'leg room' to do more then the next iteration of this process will begin with the Product Owner bringing in goals related to the next backlog, with, as said previously, the Sprint Goal only changing if the new goal has a direct relationship to it.

Believe you me, as a practicing Scrum Master this is a triage idea and far from what I would want. But if you've got yourself in this particular pickle then this has several advantages. The Product Owner is helped with the problem of prioritising across backlogs, teams have a single clear Sprint Goal so they understand the most important piece of value to deliver and it stops teams 'cramming' sprints full of work based on capacity rather than usable increment delivery.

Or just have a single backlog… it's complicated.