

Make Ready

– the missing link in Agile projects

By Bent Jensen, partner and Agile coach BestBrains aps & Sven Bertelsen, owner Strategic Counselors

The Make Ready Process can be of great use to Agile software teams. Here Bent Jensen and Sven Bertelsen give an introduction, based on their experience of Lean Construction.



An every day situation:

“Where the heck are those test data for this module?”
Sam asked.

Silence in the whole office, nobody seems to have them.

“Did we request them from the client?” he asked this time.

More silence.

“And by the way, did we order SubSoft to assist us this week? I haven’t seen them this morning. And where is the test version from HardCore on their damned new gadget?”

A sigh from the corner – but no answer.

All of a sudden question upon question comes up, and the whole developing process comes to a grinding halt. Waste of time all over the place.





Bent Jensen



Bent Jensen is Director and Consultant at the Danish Agile Consulting firm BestBrains. BestBrains primarily helps software companies implement faster and more effective ways to create software, with a Lean and Agile foundation.

Bent has worked with a number of companies on shaping their processes for distributed/outsourced development.



Sven Bertelsen

Sven Bertelsen has worked for 40 years as consulting engineer and project manager and has been partner in NIRAS for 25 years. Today, he is an external lecturer at the Danish Technical University as well as working as consultant and coach to project producing industries in productivity improvement along with research within Lean Project Production.

» This is an every day situation which we can all relate to – nothing out of the ordinary. Sometimes you may also hear conversations like this one:

“I think we should take into account that two sprints from now we are going to add x functionality. Let’s already now make sure we can easily integrate it into the architecture.”

“No – you should not think of that now. We are not supposed to do that in Agile development. Let us cross that bridge when we get there.”

PLANNING AHEAD

It is definitely true that software can be over-engineered and be configurable to handle all kinds of more or less probable enhancements in a distant future. But it is equally certain that it has never been the aim of Agile development to make teams behave more stupidly than they actually are by deliberately ignoring knowledge that is available to them and to make precautions in due time.

Teams with this approach are faithful to the letter of Agile methods but not to their intent. They can be compared to a group of scouts going on a hike who, when they start, know they will have to cross a river some time later on the trip. However, when one guy wants to bring some rope the others tell him to leave it at home, because “when we get there, we will find a way to cross the river”.

Anti-planning Syndrome

More of something good is not always better. That holds true for most principles in software engineering. “The simplest thing that can possibly work” does not mean “Ignore everything you know” but rather “Do the simplest thing that may possibly work” (in the context of your current knowledge).

Overdoing this principle can lead to the “Anti-planning Syndrome” which

can sometimes be seen in Agile development. We have recognized that Big DETAILED design upfront BDDUF is not good, because we can’t oversee the complex nature of the problem anyway. But that does not mean we should not spend enough time to understand the core of the problem and design at least the outline of a solution we are going to apply, before diving in and starting to write code.

Doing it right

So how much planning is optimal? One way to look at it is this: if your team are routinely modifying what has been planned in detail, then you are planning too much, too early. If, on the other hand, they routinely have to redo work or get stuck halfway through a task because they miss information, decisions or deliveries from others, you are planning – or rather preparing – too little.

There has been a lot of emphasis on “Done”-criteria and making work really “done-done” in the Agile team process. The other side of the process, “Only working on stuff that is really ready”, has not gained the same attention, even though the consequences of starting work which is not ready is at least as costly as not delivering work which is not really done.

Putting backlog grooming to work

To avoid the waste of starting work that cannot be completed, you may have to implement a process that will ensure that only ready work is undertaken by the development team. On a number of occasions we have implemented the Make Ready Process in Agile teams, to help them bring the stuff they need to cross the river.

The Make Ready Process consists of a process similar to the development process in Scrum and other Agile processes. Make Ready starts with loosely

defined user stories or other forms of functional requirements and ends with user stories ready to work on. The process is aimed at giving maximum speed in development, so when a user story is actually ready to be started, it is most likely to be completed in the current sprint.

Guidelines show the way

The criteria for when a user story or a task is ready to work on, will differ from project to project. However some guidelines can be found in looking along the streams which feed the flow of work and make the work packages “sound”. By a sound work package we mean a task that is in good shape regarding:

Dependencies

Is there any development work that needs to be completed before this should be done? Do we need to coordinate with other teams?

Skills

Do we have the skills on the team to complete the task?

Space

Can we actually get to make the change. What is the state of the code we are looking at?

U.X designs

Does the User Story have an associated U.I design and workflow?

Stakeholders

Is there anyone that needs to be informed and involved in this user story that maybe later can disapprove?

Work ahead

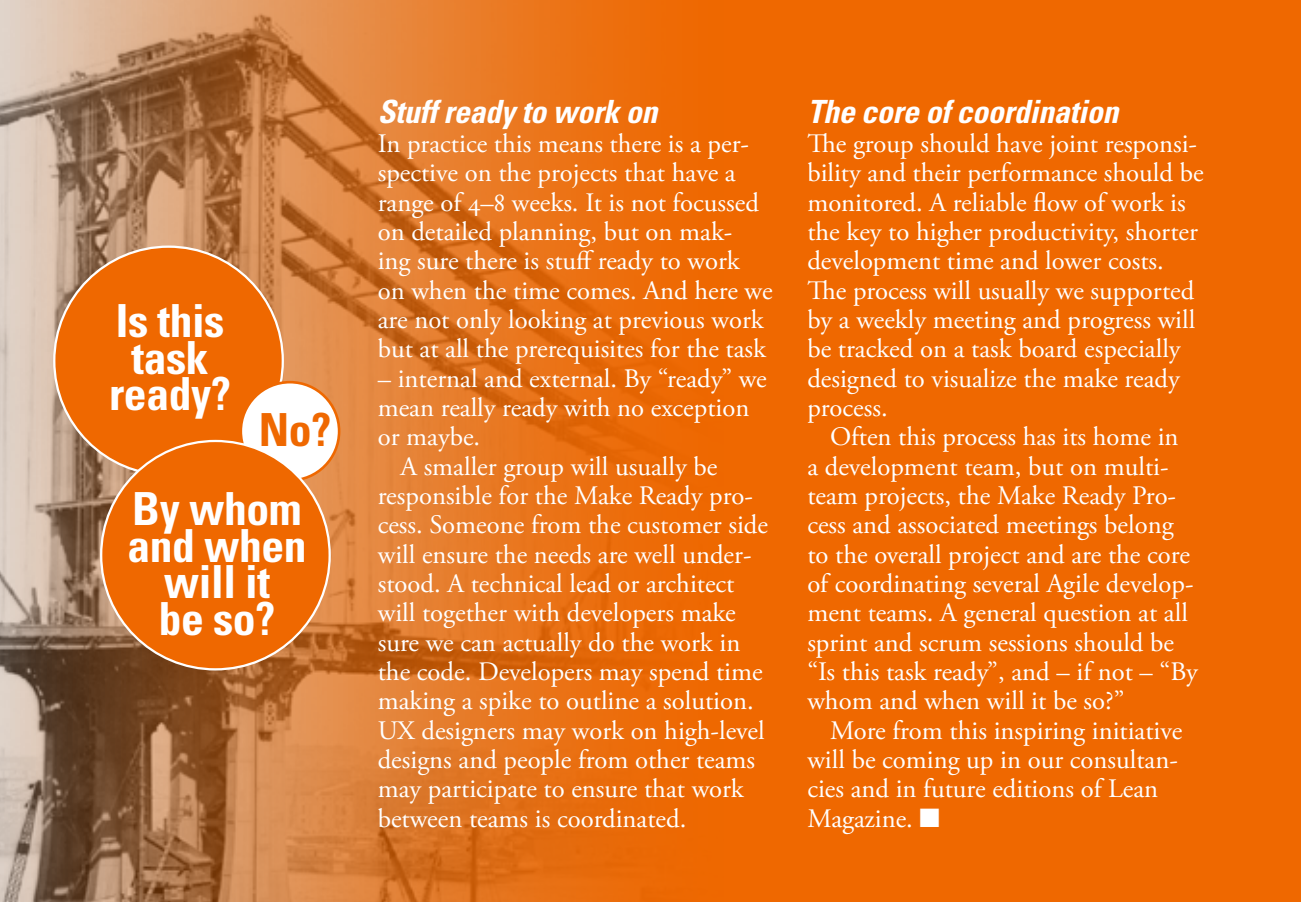
Is there anything coming later that this development work needs to take into account?

Following work

Are the receivers of this work in place and ready to receive it?

In construction this process looks at:

- Previous work
- Space
- Crew
- Equipment
- Information
- Materials
- External conditions i.e. the weather, clients' OK, financing, and authorities' approval.



Is this task ready?

No?

By whom and when will it be so?

Stuff ready to work on

In practice this means there is a perspective on the projects that have a range of 4–8 weeks. It is not focussed on detailed planning, but on making sure there is stuff ready to work on when the time comes. And here we are not only looking at previous work but at all the prerequisites for the task – internal and external. By “ready” we mean really ready with no exception or maybe.

A smaller group will usually be responsible for the Make Ready process. Someone from the customer side will ensure the needs are well understood. A technical lead or architect will together with developers make sure we can actually do the work in the code. Developers may spend time making a spike to outline a solution. UX designers may work on high-level designs and people from other teams may participate to ensure that work between teams is coordinated.

The core of coordination

The group should have joint responsibility and their performance should be monitored. A reliable flow of work is the key to higher productivity, shorter development time and lower costs. The process will usually be supported by a weekly meeting and progress will be tracked on a task board especially designed to visualize the make ready process.

Often this process has its home in a development team, but on multi-team projects, the Make Ready Process and associated meetings belong to the overall project and are the core of coordinating several Agile development teams. A general question at all sprint and scrum sessions should be “Is this task ready”, and – if not – “By whom and when will it be so?”

More from this inspiring initiative will be coming up in our consultancies and in future editions of Lean Magazine. ■



LEARNING FROM LEAN CONSTRUCTION

We have looked deeply into Lean Construction, a new way of understanding and managing projects, which is getting increasingly more attention in the construction industry all over the world. Again and again more than twenty percent increase in productivity and speed of delivery are reported along with fewer faults and increased earnings all around. These remarkable results are not found in construction only but also in shipbuilding,

not least in Norway. Our lesson has been that we are not alone with our problems, and some of them have been solved and solutions demonstrated in practice for at least a dozen years. And even better: most of the solutions are surprisingly simple, as you may see in the introduction to the Make Ready Process.

This article is based upon an ongoing management developing project, which we have initiated in co-operation with the Danish consulting firm: Best Brains and with the assistance of Sven Bertelsen. Sven is an highly experienced project manager who has for the last ten years researched and advised within the field of Lean project production where he is today recognized as an international expert. ■