

Introducing Scrum for Hardware and Systems Design into Your Organization: A Checklist

Have you been thinking about trying Scrum with your hardware and systems design teams? Here is a checklist that will help you evaluate if you have all the necessary pieces in place.

The Scrum methodology has had a significant impact on the design of software. It has cut the number of failed projects nearly in half. Can it have the same impact on your hardware and systems teams? It might, but certain factors need to be in place to maximize your chances of success. The twelve questions below address these factors as a checklist. If you can tick all the boxes on the list, then you stand a chance at revolutionizing your product design process. Empty boxes point to areas that need addressing before you try to move to Scrum.

SCRUM FOR HARDWARE AND SYSTEMS CHECK LIST

- Are the expectations for Scrum introduction realistic and agreed upon?
- Are there Scrum champions and an executive sponsor?
- Are the sponsors and champions aware of the organizational impediments?
- Do the leaders and the team have adequate training and coaching?**
- Is there an identified first project?
- Is there a team with the right skills to tackle this project?
- Do the team members have dedicated, concurrent time for the project?
- Is there a Scrum Master on the team who is strong enough to keep the team from being interrupted and has sufficient Scrum training to guide them?
- Is there a Product Owner on the team who is empowered to represent the voice of the customer?
- Is there a sense of the uncertainty in the specifications and the technologies being considered?
- Does the team have Test Driven Design attitude and have provisions for rapidly prototyping and testing artifacts?
- Is there good communication with teams responsible for other systems that interface with system being designed?

Developed by David G. Ullman, ullman@davidullman.com,
www.mechdesignprocess.com

[Attend our 3 hour Introduction to Scrum Course, Nov 12, 2020: Noon to 3:00 PST.](#)
[Click here](#)

1. Are expectations realistic and agreed upon?

Organizations consider using Scrum for a variety of reasons. A German research group asked Agile practitioners to rate their reasons for adopting it. Most all the "Agile" respondents were using Scrum, so these results apply to Scrum. The highest-ranked of the 26 reasons are listed in Table 1. Also shown are the actual benefits.

Key results are:

- Even though the most important reason for adopting Agile methods is to "accelerate product delivery," the highest benefit was to "manage change" This benefit highlights Scrum's importance during new product development when uncertainty and change are important.
- While few in the survey thought "project visibility" was an important reason for adopting Scrum, it is the second most beneficial. Scrum is a method to improve communication both within the team and beyond it. Visibility begets communication, so this result is not a surprise.
- "Increase productivity" was both an important reason for adoption and a significant result.
- The top six benefits in order are:
 - Manage change
 - Improved project visibility
 - Better business/engineering alignment
 - Increased productivity
 - Accelerated product delivery
 - Improved team morale

Measures	Reason for adoption Agile	Benefits of adopting Agile
Accelerate product delivery	75%	61%
Manage change	64%	71%
Increase productivity	55%	62%
Better business/engineering alignment	49%	65%
Increased product quality	46%	47%
Enhanced delivery predictability	46%	49%
Improve project visibility	42%	66%
Reduce project risk	37%	47%
Improve team morale	28%	61%

100% = great value, 75% = strong value, 50% = some value, 25% weak value, 0% no value

What is essential before considering Scrum in your organization is that the reasons for trying it are clear, agreed upon, and measurable. If changing from a pure Stage-Gate or another framework to Scrum, first note how well the current method performs, so there are results to show that it improves the situation.

Further, the 2019 Chaos Report showed that when using Agile (e.g., Scrum) methods, the number of successful software projects – those on schedule and with the required

features - nearly doubled. You need to understand how many of your current projects are successful (at least subjectively) to know if Scrum helps.



2. Are there Scrum champions and an executive sponsor?

Generally, Scrum champions are product managers working with a successful software Scrum team or have read of Scrum's success. This "champion" then tries to get Scrum going in their group. While a champion is needed, having only one in middle-management is fragile as, if the champion cannot show early success and get executive support, or they leave the group, the effort is doomed.

An alternate approach begins with an executive sponsor who wants to improve some of the process measures listed in Table 1. They become both sponsors and champions. One company I consult with has had success with Scrum in software and has middle-level management champions. Still, there is not an executive sponsor for integrating Scrum beyond software development. Scrum isn't going to happen here.

In another company, the engineering director is the sponsor/champion, and Scrum shows significant benefits. They were just enabling Scrum when the pandemic sent their design teams home. They credit Scrum with enabling the now distributed online team and have introduced new products since March using it.

3. Are the sponsors and champions aware of the organizational impediments?

All organizations have built-in impediments to change. This is well stated as "Drucker's "Cuckoo Effect." The questions are, how strong and, even more importantly, how well known are the antibodies? One group I work with is a family built, owned, and managed corporation. The antibodies are well known and very strong. Change happens very slowly here.

"Any innovation in a corporation will stimulate the corporate immune system to create antibodies that will try to destroy it"
Drucker's "Cuckoo Effect"

4. Do the leaders and the team have adequate training and coaching?

While there are many books and online Scrum courses, it is essential to have sufficient training and coaching, at least initially. Further, virtually all the training is by and for software developers. As I wrote about in an earlier Machine Design (Dec 27, 2019) article,

"13 Challenges When Applying Scrum to Hardware Design,"¹ there are significant differences between designing software and designing hardware and systems. Someone with a mature knowledge of hardware and systems design processes should be helping you get started.

5. Is there an identified first project?

A team should not try to do Scrum half-way. A new project should be started from the beginning. It is the only way to measure the difference between the previous and the new one and not be saddled with prior decisions.

6. Is there a team with the right skills to tackle this project?

There are three roles on a Scrum team; the Product Owner, the Scrum Master, and the technical team. The Scrum Master must understand the Scrum process or have advisors to help them do that. The Product Owner must understand the need for the product and is the team-side voice of the customer. The technical team must have the needed technical skills to develop the product.

If all these roles are not in place, then the potential for success is low.

7. Do the team members have dedicated, concurrent time for the project?

A crucial part of Scrum is dedicating time for communication amongst the team members. Ideally, they should all be working on the project at the same time in a face-to-face environment. While this has become more difficult in 2020, they still need dedicated, non-interrupted time for the project even if not collocated. See the next measure for more on Scrum's dedicated time management.

In fact, the Scrum structure strongly supports online collaboration. Using an online Scrum board like Jira or Trello gives project visibility and agility that is essential.

8. Is the Scrum Master strong enough to keep the team from being interrupted and to guide them?

Typically, engineers have many small tasks to do even when focused on a larger core project and may often be interrupted with "fires" that need fighting. A company I was working with to introduce Scrum had many "fire" issues. What I suggested (and they implemented with success) is that they devote a focused three days a week to the Scrum project and use the other two days as a buffer to do all the other needed work. During the three days of focus, the Scrum Master's responsibility was to deflect any interruption to the buffer. This structure has worked well for this organization. In fact, focusing the small stuff in the buffer time has actually reduced the needed time for it to less than two days, leaving more time to focus on the core project.

¹ <https://www.machinedesign.com/community/article/21119382/13-challenges-when-applying-scrum-to-hardware-design>

Or the video version

bit.ly/13scrumchallenges

The Scrum Master must:

- Enable the process. The Scrum Master must be knowledgeable about Scrum methods and see to it that they are followed. The Scrum Master does not have any say on the actual design process that the team follows in their day-to-day activities, only that whatever they do must adhere to the flow described in this chapter.
- Facilitate meetings.
- Maintain situational awareness of the work.
- Remove obstacles and protects the team from interference.
- Manage people and processes. The Scrum Master is the closest role to a manager in the traditional sense. Where traditional managers would try to control the team, the SM focuses on protecting the team from interruptions and removing obstacles.

9. Is the Product Owner empowered to represent the voice of the customer?

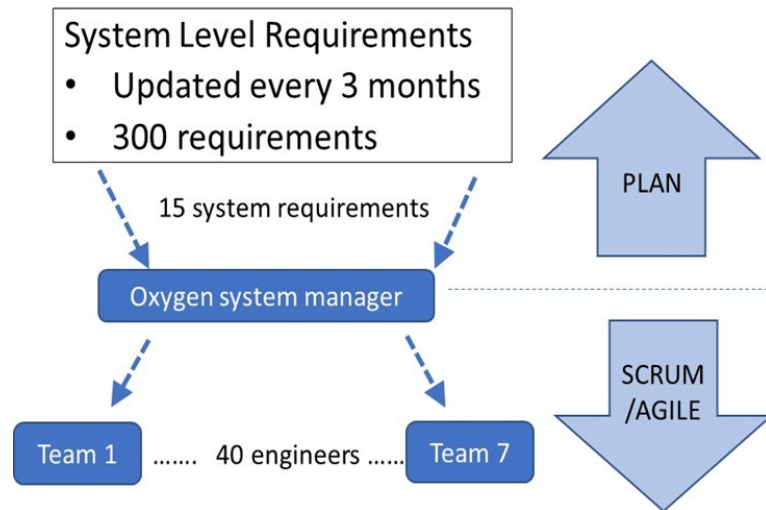
The Product Owner (PO) is the team's face to the outside world. They define the specifications and decide on the sequence of deliverables. To fill this role, the PO must be:

- Knowledgeable about the domain. This is not to say that the PO must be an expert in any specific product function or have knowledge about all the functions but must have a general understanding about the technologies that might be used during its design and manufacture. Additionally, they must be knowledgeable about the market for the final product as they are the conduit between the team and the voice of the customer. The PO is often a marketing person.
- Empowered to make decisions.
- Available to the team to explain what needs to be done and why.
- Accountable for product value.

10. Is there a sense of the uncertainty in the requirements and the product?

This question's meat is the evolution and management of product requirements (aka, specifications, constraints, or customer needs). At one extreme are projects with no written requirements, and at the other end are projects where the first gate is a complete set of requirements needed before any concepts are developed. The middle ground is seen in Saab during the development of the Gripen E jet fighter.

For the entire fighter, there are about 300 requirements, representing the demands of Saab's customers. These were developed upfront and are reviewed every three months and updated as needed. Of these, about 15 define the need for the oxygen system to provide life-sustaining air in flight and after ejection. This set of macro-requirements is all that is provided to forty engineers on seven teams who do the design work and who develop the micro-requirements, those that are important only to the oxygen team and not to the other systems. This structure forces the decision-making down to the team level making the development process very efficient and agile.



11. Does the team have provisions for rapidly prototyping and testing artifacts?

One goal of Scrum for software is having deliverable code at the end of each 2-4 week development sprint so the customer can use and effectively test the result. In reality, this seldom occurs for software, and for physical systems, this is nearly impossible. Further, software developers usually write function testing code within the code, something also not often possible with physical systems.

That said, the goal can be relaxed to emphasize test-driven development (TDD). TDD means that when creating a product development task, also generate the test the part, module, or assembly is expected to pass. This test may be as basic as counting the number of concepts developed or as complex as statistical results from the lab.

Focusing on TDD encourages creating the most uncomplicated design possible that enables passing the test. This philosophy forces emphasis on the deliverables rather than the activity- product over action.

12. Is there good communication with teams who interface with the Scrum team?

One of the worst mistakes in product design is when one team makes a change, and this change is not communicated to another team whose system is affected by it. There are engineering horror stories of parts not fitting or not functioning because the communication across an interface broke down.

Scrum encourages modularity, which is natural in writing code. For physical systems, it forces early attention to product architecture and of modules (i.e., assemblies or systems) with fixed, stable interfaces. This type of architecture is most immune to uncertainty and changes as the product matures. Stable - fixed interfaces must be

consciously designed into the product. Designing from interfaces to parts and assemblies is good design practice as championed in my text "The Mechanical Design Process." Designing from interfaces helps uncouple complex systems and increases communication between design teams. Scrum encourages this practice.

Summary

This checklist is a good starting place for Scrum's success when designing hardware and systems. There is much detail in the Scrum process that has evolved to support these twelve. The transition to Scrum is not easy, but the potential benefits are large. Using this checklist to assess your readiness is a good starting place.

For an introductory background to Scrum, see the Machine Design article "Scrum for Hardware and Systems Design" in the May 29, 2019 issue.

<https://www.machinedesign.com/3d-printing-cad/article/21837829/scrum-for-hardware-and-systems-development>

To help you understand how hardware and systems design presents challenges not had in software, read "13 Challenges When Applying Scrum to Hardware Design" in our Dec 27, 2019 issue.

<https://www.machinedesign.com/community/article/21119382/13-challenges-when-applying-scrum-to-hardware-design>

For complete coverage of Scrum for hardware and systems, see

www.mechdesignprocess.com/landingpro