Construx®

5 Common Gaps in Scrum Adoptions

Jenny Stuart
Vice President of Consulting
Jenny.stuart@construx.com
www.construx.com

Copyright Notice

You Know Scrum

Scrum is the most widely used Agile framework in the software development industry today.

One of Scrum's advantages is that it provides a significant amount of structure: specific roles, events, and artifacts that work together.

But Do You Know the Common Gaps in Scrum Adoption?

Construx works with many organizations across a diverse set of industries.

We find that many teams encounter a consistent set of challenges with Scrum due to similar gaps in their Scrum adoptions.

This webinar describes those gaps and shares practices that will help you address them.

Construx®

Scrum Adoption Gap #1:

Missing or Insufficient Focus on Quality

Scrum's Definition of Done

The Scrum Guide states that Scrum Team members must have a shared Definition of Done (DoD). However, it provides no specific and concrete guidance on what should be in your DoD.

The Scrum Guide is intentionally vague because the details do—and should!—vary widely from team to team.

Scrum's Definition of Done

Many of the Scrum teams that Construx works with do not have a shared DoD that everyone understands, agrees to, and uses.

Instead, we see a series of antipatterns.

Definition of Done Antipatterns

Missing Definition of Done Many teams never establish a DoD.

Ignored Definition of Done The team has a definition but doesn't use it during the Sprint to determine whether an item is ready for the Sprint Review.

Weak Definition of Done The DoD omits quality or addresses it superficially. For example, development is considered complete without unit and functional testing, documentation, and other elements necessary for high-quality work.

A mandate from on high Someone somewhere in the organization provides a DoD without consulting with the teams. The imposed DoD contains inappropriate items and items that can't be completed within the Sprint.

How to Use a Definition of Done

A DoD should state the teams' expectations for the following:

Testing New unit, integration, functional, and system testing. May include running selected manual regression tests. Often includes automating some of all of the test cases.

Reviews Code reviews and test cases reviews.

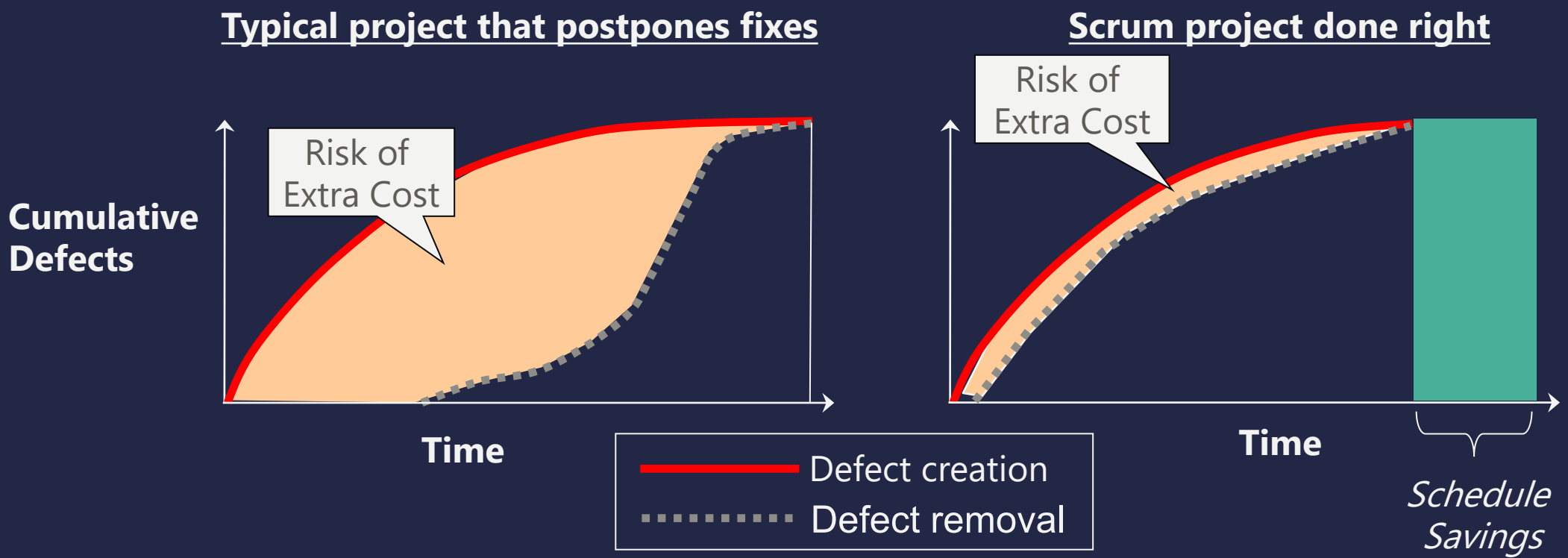
Tool use Static code analysis, security checks, and code style checks.

Documentation Updating architecture specifications, design documentation, user documentation, release notes.

Deployment Deployment scripts or guidance.

The Hidden Cost of Defects

A project that allows defects to build up (whether visible or not, e.g., **technical debt**) increases uncertainty (risk) while eroding the capability to deal with it.



Construx®

Scrum Adoption Gap #2:

Insufficient Staffing of the Scrum Roles

Scrum's Roles

Scrum defines three roles: Product Owner, Scrum Master, and Team Member.

Each of these roles has a unique set of defined responsibilities that complement the other roles' responsibilities. In combination, these three roles provide checks and balances.

Insufficient staffing of the Scrum roles is a common, and painful, gap.

Scrum Role Antipatterns

Missing Product Owner A team without a PO lacks sufficient guidance on what the business and stakeholders need. Because the PO guides the work of a team of up to nine people, it is a vital role.

Inactive Product Owner The PO but doesn't participate with the team sufficiently, if at all. The team can identify the PO but doesn't benefit from this.

Proxy Product Owner The right person doesn't have the time to be, or interest in being, a PO, so someone else is designated as Proxy PO. Proxy PO often cannot provide the detail and day-to-day information necessary for Scrum to be successful.

Scrum Role Antipatterns

The "oh by the way" Scrum Master Many teams have a Scrum Master who was the person most willing (or least unwilling) to take on the role. Sometimes this is better than having no Scrum Master, but often these people aren't interested in becoming a great Scrum Master. A great Scrum Master brings much more just facilitating the Scrum events..

The "command and control" Scrum Master This Scrum Master tells the team what to do rather than enabling it to become a self-empowered, thereby greatly impeding learning.

Make Sure to Staff the Scrum Roles

For Scrum to be successful, it is crucial to staff the Product Owner and Scrum Master roles appropriately.

Organizations are rarely able to hire new staff when they transition to Scrum. Instead, it is a matter of identifying how the organization can best fill each role by using its existing people.

Construx®

Scrum Adoption Gap #3:

Insufficient Backlog Refinement

Backlog Refinement in Scrum

The Scrum Guide states, “Product Backlog refinement is the act of adding detail, estimates, and order to items in the Product Backlog,” and it says that this process should consume no more than 10% of the team’s time on an ongoing basis.

Although backlog refinement is not a defined event in Scrum because there is no set time in the Sprint at which it must occur, it is crucial for successful Scrum work.

Insufficient Backlog Refinement Symptoms

Sprint Planning takes longer than necessary Good refining ensures Sprint Planning is a focused, efficient, and effective event. Sprint Planning should focus on the *how* we will build it because the *what* we're building has been sufficiently understood in refining.

Work consistently slips into the next Sprint Many teams bring in stories that are too large to be brought to their Definition of Done within a Sprint. Instead, sufficient backlog refinement must be done for large stories to be decomposed into items that are small enough to fit in a Sprint.

Insufficient Backlog Refinement Symptoms

There are excessive questions about the details of the work during the Sprint Some teams are unable to complete work in a Sprint because they do not understand important details of what they're building.

The team experiences excessive downstream work When lots of required changes are identified in the Sprint Review or after the Sprint is complete, it is a sign that the work was not well understood by the entire team.

Make Sure to Perform Thorough Backlog Refinement

Construx encounters many teams that could vastly improve their performance by reducing easily preventable rework through better conversations and backlog refinement.

A good starting point is to spent one hour twice a week refining the Product Backlog. More shorter sessions provides the best balance of flexibility and time to focus on the task.

Construx®

Scrum Adoption Gap #4:

Missing or Ineffective Retrospectives

Sprint Retrospectives

The Sprint Retrospective is a process and teamwork inspection event—a powerful tool that can make Scrum teams great.

But many teams' retrospectives do not generate significant improvement. Some teams stop holding them when they discover, correctly, that poor retrospectives are a waste of time.

Common Sprint Retrospective Issues

Failing to address core issues Important issues are never exposed, discussed, or addressed because of cultural barriers, interpersonal issues, organizational politics, line management reporting conflicts, or other reasons.

Failing to identify a change the team will make in the next Sprint Holding a retrospective without identifying an improvement misses the entire purpose of the retrospective. It isn't sufficient to talk about what worked and what didn't.

Agreeing to a laundry list of issues Teams should identify one valuable change, perhaps two, that they will make in the next Sprint. Agreeing to a long list of changes generally means that no change will be made.

Common Sprint Retrospective Issues

Identifying vague improvements Actions such as "We should do better refining" or "We should meet our Sprint commitments more often" are hard to implement because they are unmeasurable. Teams should instead brainstorm specific, measurable changes that they can try so as to improve.

Holding the same retrospective over and over and over The most common technique is asking people what worked and what didn't work, using the "go around the table" approach. Any technique used repeatedly quickly becomes boring and is unlikely to generate unique, valuable insights.

Common Sprint Retrospective Issues

Not looking at past performance It's useful to mine the team's historical data and share it in retrospectives to help the team see patterns. Data like the three-Sprint moving average velocity, initial total hour estimates vs. actual, and % of Product Backlog Items committed vs. delivered can be illuminating.

Never or rarely making the changes identified in the retrospective An identified change can be impactful only if the change is made. Teams can add action items related to the change to the backlog so that they are visible and any work necessary to implement the change can be estimated and resourced.

Make Your Retrospectives Useful

Retrospectives should be fun and energizing.

Retrospectives should result in continuous improvement via Scrum's *inspect and adapt* mantra.

Construx®

Scrum Adoption Gap #5:

Lack of Focus on Incremental Value Delivery

Incremental Value Delivery

Many teams throughout the industry use Scrum, but few of them create potentially shippable software at the end of each Sprint.

Teams that do create shippable software for each Sprint often have disconnects with their stakeholders that have prevented the team from delivering what is most valuable first.

Incremental Value Delivery

Scrum places a lot of responsibility on the Product Owner:

- Understand the team's stakeholders.
- Discover the stakeholders' needs.
- Properly order the Product Backlog to bring the highest return on investment to the business.
- Convey the details to the team so that they can quickly and continuously deliver stakeholder value.

Common Problems that Prevent Incremental Value

Construx often sees these problems:

- Missing key stakeholders.
- Failing to deeply understand what the stakeholders value.
- Not staying current with changing stakeholder values.
- Not exposing key assumptions in decision making and prioritization.
- Breaking large pieces of work down in ways that don't deliver value incrementally.

Make Sure to Focus on Incremental Value

Scrum is an excellent delivery engine, when used properly.

The Product Owner and Product Backlog are the steering mechanism for that engine.

A significant amount of work is necessary to ensure that the right things are placed at the top of the backlog.

Resources Available from Construx

Construx provides free resources to help individuals, teams, and organizations grow their software development skills:

https://www.construx.com/resources/topics/

Check out our “Inspect & Adapt” podcast for more information on software development:

https://www.construx.com/resources/podcast/

Construx®

We believe every software team can be more successful. Construx is led by *Code Complete* author Steve McConnell, and our team is comprised of professionals who are software experts first, software trainers and consultants second. Our mission is to make your software teams more effective.

For information about our training and consulting services, contact info@construx.com.

www.construx.com

10900 NE 8th Street, Suite 1300

Bellevue, WA 98004

+1 (866) 296-6300