TESTING AND QUALITY IN SAFE

The Scaled Agile Framework for Lean Enterprises (Safe) is the most popular framework among large organizations to achieve business agility. In this article Mette Bruhn-Pedersen and Derk-Jan de Grood explain how testers can contribute in a Safe environment.

Mette Bruhn-Pedersen
Derk-Jan de Grood

Organizations are increasingly starting to understand that business agility and responsiveness are key factors to survive and stay ahead of their competitors. To yield value, the work of individual agile teams should therefore be embedded in larger business processes. Many organizations embrace the Scaled Agile Framework for Lean Enterprises (Safe) to enable multiple teams to collaborate on a single release, to plan and manage dependencies and to translate strategic needs into user stories that can be completed by individual teams.

Safe is a freely revealed knowledge base of integrated, proven patterns for enterprise Agile-Lean development. Among the available scaling frameworks it seems to be the most thoroughly documented. Other frameworks such as Less and Nexus are less well documented and also less prescriptive. This might explain Safe's popularity with organizations that are used to formal processes and want a clear structure.

To illustrate our points we'll look at Safe's Portfolio configuration, which has three levels. At the bottom there's the team level. This describes practices for individual and cooperating teams working on user stories. It builds on familiar Scrum and Kanban practices. One step up, the programme level describes how organizations divide work among individual teams and how they merge completed features into a continuous delivery pipeline. The agile release train bundles the teams' work into controlled product increments. At the top, the portfolio level translates strategic organization themes into value streams and defines portfolio epics.

Quality at the team level

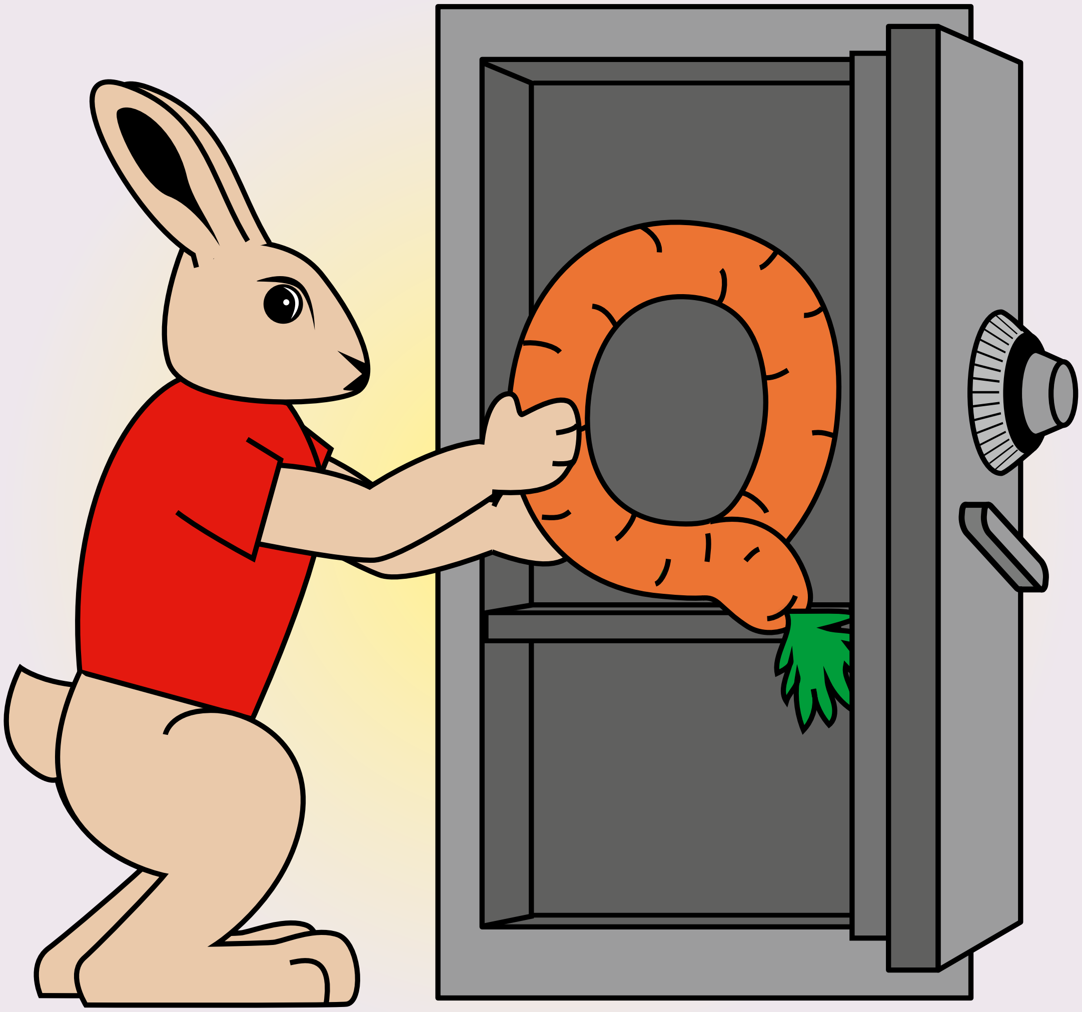
One of Safe's core values is that quality is built in. Built-in quality practices ensure that each solution element, at every increment, meets appropriate quality standards throughout development. That's a good starting point.

The test-first practice recommends building tests before writing code. Another Safe practice is pair work. At the team level testers can pair with both users and developers to co-create tests. Testers lacking coding skills can still pair with team members to review the automated tests and discuss the frequency with which tests should be run. By taking a leading role testers can ensure that all team members learn and use agile testing methods. This practice gives the development team a good understanding of the problems to be solved, both functional and non-functional.

On a more practical level, testers can review the Definition of Done (DOD) to help teams define quality measures. The DOD is a great tool for embedding quality into the process. During discussion team members are likely to identify valuable tests that aren't being done or are done outside the sprint. Comparing the DOD used by one team with those of the other teams enables cross-team alignment. It's really helpful for Agile coaches and the Safe programme consultant if test advocates contribute their vision and knowledge on how to embed these quality measures into agile processes.

In practice we often see that tests are still executed manually. Automated tests (if any) are usually created after coding instead of before, and are often run from a separate platform not integrated into the build process. The release process is often only partly automated, and therefore vulnerable. Safe states that organizations should aim for a repetitive and hands-off build process enabling quick deployments in various environments. Testers can contribute by emphasizing the need for this and ensuring the pipeline includes automated functional and non-functional regression testing.

Unfortunately, Safe does not describe these practices in detail. Neither does the framework talk much about formal roles in testing. It doesn't explicitly address the need for an effective quality strategy to



support the agile teams. We think all test professionals can contribute at the team level by calling attention to quality practices and the need for a quality strategy, and by helping to implement relevant quality assurance and testing practices.

Quality at the programme level

The quality strategy will most likely be defined on the programme level, since it outlines what's needed to deliver an integrated and tested solution to customers. Its implementation will also affect the team level. A quality strategy ensures that the critical aspects are clear from various perspectives: business, technological and operational. It outlines what needs to be tested, and how this testing should be done. It may be effective to include views and interests from external suppliers and stakeholders such as compliance officers.

A quality strategy should also define how feedback on product quality and progress is gathered. Taking the programme increment (PI) objectives as a basis, the quality strategy can outline concrete ways to collect feedback. Such a strategy should also address test quality. Does the test work need auditing? How will the teams be coached on their testing? Another item to address in the quality strategy is how to conduct tests that don't fit into a sprint.

In practice we all too often see organizations that lack both overview and focus. A clear quality strategy that aligns teams and their work and provides shared insight into the work to be done not only leads to better-quality solutions, but also reduces the number of late surprises and enables teams to discuss progress and impediments and to replan their roadmap if necessary. It ensures that teams know what's expected from them and that testing is not forgotten, and it provides room for coaching and teaching.

That last one is a key issue. The adoption of Agile makes testing a team responsibility, and more often than not it's done by developers and users with limited test expertise. They feel uncertain about the way they test, or they lack enthusiasm because they don't know what to do. They can do an even better job if they receive relevant training and are helped by experienced testers to improve test design, execution, tool support, logging and reporting. This guidance process will give organizations a multidisciplinary view of quality and will increase team flexibility.

In Safe the last sprint in the PI is reserved for PI planning and (if needed) the integration of the various system or solution assets. Testers can typically contribute here by facilitating risk analysis and root cause analysis (RCA). Identifying business and technical risks prior to or during PI planning may lead to extra acceptance criteria or even new user stories. Many organizations fail to regularly perform an RCA and thus fail to understand the relationship between incidents and testing. By setting up a feedback loop testers can help to build the right solutions and ensure that the agile teams know what tests matter the most.

The programme level in Safe is also designed to help alleviate typical integration challenges. End-to-end integration should be a starting point for planning development and cross-team collaboration. Testers and test managers should emphasize a quality mindset so that integration testing is taken into account during PI planning. Integration tests should be executed in each iteration, but some tests might be better suited for the PI's final sprint.

To get feedback and learn how the system is used, it's important to deploy frequently and with as few delays as possible. Testing the operation model to ensure operational readiness should be addressed while defining the PI objectives. Integrating the business-readiness testing in the overall quality strategy enables an early time-to-market.

Another challenge within Safe is governance. Progress and quality are often discussed at a scrum-of-scrums meeting, but progress indications are often subjective and incomplete. Testing helps provide transparent and objective insight into the available working software. The availability of lean, mean but adequate data to assess release progress enables better planning and makes it easier to revise the release train roadmap.

Quality at the portfolio level

The portfolio level is the linking pin between organizational goals and development work. It addresses the what and the how, across both technical and cultural aspects. If we want to embed quality into the organization, we should do it here.

Safe doesn't define a quality ambassador at the portfolio level, but it might be very useful to have one. He or she can see to it that strategic themes don't focus solely on 'new business functionality'; technical debt reduction, test architecture, and developing and supporting good QA practices must be prioritized as well since they enable a sustainable delivery rate.

A portfolio-level quality ambassador can also be instrumental in prioritizing compliance epics. Compliance isn't often second nature to IT people, leading to late surprises and penalties. Defining compliance epics and implementing the required security, traceability and revenue assurance functionality at the portfolio level ensures that the right proof (logging, testing of controls and documentation) is available. This proof will then be evident during audits and will reduce the misalignment between what the authorities need and what's been implemented.

Safe describes the quality management system (QMS) as a set of approved practices, policies and procedures. It ensures that development activities and outcomes comply with all applicable regulations and provides the required documentation to prove it. The QMS might be part of the compliance officer's portfolio, but the quality ambassador should also include it in the quality strategy.

The portfolio level is all about business. Organizational readiness at the programme level can be a way to ensure that technical solutions will be used and yield benefit. An assessment is advisable here: do the implemented strategic themes deliver the expected value? The portfolio-level quality ambassador can help the organization define KPIs that provide this insight and help to broadly assess the user experience of the value streams.

Quality at all levels

Communication and dependency management are typical challenges in scaling. The measures we've discussed focus on having a strategy so people know what's expected of them, taking quality into account during planning sessions and ensuring that quality-related work gets sufficient priority. We find there's a need for a quality focus at all levels and recommend taking a strategic approach to quality and testing.

People who currently work in testing and quality have a wealth of knowledge and can take on an ambassador's role to promote built-in quality. Test professionals may need to do some upskilling. Automating tests and implementing a continuous delivery pipeline require technical knowledge at the team level; to be effective at the programme and portfolio levels also requires business skills. To make Safe effective, test professionals can help others build in quality and facilitate broad collaboration. This means non-testing professionals will skill up on quality and testing practices. We can help them through coaching, teaching and showing them the strategic advantages a quality focus confers.

Scaling requires everyone to collaborate to make it work. Testers might very well take the lead in creating the awareness that true business agility requires built-in quality, and that it's not enough to implement just Safe's current guidance on quality practices. It requires attention on every level, discipline and quality ambassadors ready to relentlessly help build quality solutions.

Mette Bruhn-Pedersen works as a tester, test manager, test lead and agile transformation coach primarily in the financial sector at her company Safe Journey. Derk-Jan de Grood works for Valori as a senior test manager and agile transition coach. In 2016 he published the book 'Agile in de echte wereld – Starten met Scrum' ('Agile in the real world – Getting started with Scrum').

Edited by Nieke Roos