Don’t let frameworks distract you from the real goals!

labs.sogeti.com/dont-let-frameworks-distract-you-from-the-real-goals/

September 27, 2019

BY : Leo van der Aalst

In an attempt to implement Agile, many organizations have entangled themselves in the web of frameworks such as Scrum, Kanban, SAFe, LeSS, Spotify, and so on. Tremendous time and money is invested in this, and while we see some results, I believe we are forgetting the main purpose of these implementations — to design, build, run and maintain a valuable high-quality software. Don’t you agree? The focus today is so much on getting the frameworks right that we forget the underlying motto of building the quality software.

But I believe it’s time to take back the reigns and not let the ‘non-software development’ frameworks distract us from the real goals. And the best way to do this is to use an approach, what I have called, Quality Mapping (QM).

A Step Back
Before diving into QM, let’s take a closer look at Agile and frameworks. For me, the phrase ‘working Agile’ is synonymous with the software development done with an Agile mindset. The latter is the mindset as defined in the four values and twelve principles of the Agile Manifesto. [https://agilemanifesto.org/]

As I see it, a mindset is framework independent. So, you could work with an Agile mindset in all of the Agile frameworks or even in the more traditional, or sequential, development environment.

With a mindset to ‘*deliver valuable high-quality increments of working software in time-boxed short iterations, through adaptive change, as more information comes to light in a communicative and collaborative manner’*, you really don’t need a framework. Of course, you might use it, but you could also argue that people with an Agile mindset together with a proven software development approach (for example, DSDM) is enough to build a valuable high-quality software.

Another issue is that, in spite of a few frameworks’ focus, the Quality Assurance & Testing (QA&T) activities are often not well explained or integrated with the frameworks. In such cases, if you do use a framework, you can request a quality architect to fill in the blank spots or you use a QM approach.

You can use the QM approach with every framework, because QM is as independent of the used framework as working with an Agile mindset.

But, what is QM?
QM is an effort of each person involved in designing, building, running and maintaining software to be *transparent* on how the person gives substance to six pre-defined Quality key areas[1]: QA Awareness, QA & Testing, Governance, Transparency, Automation and Infrastructure.

How does the QM approach work?

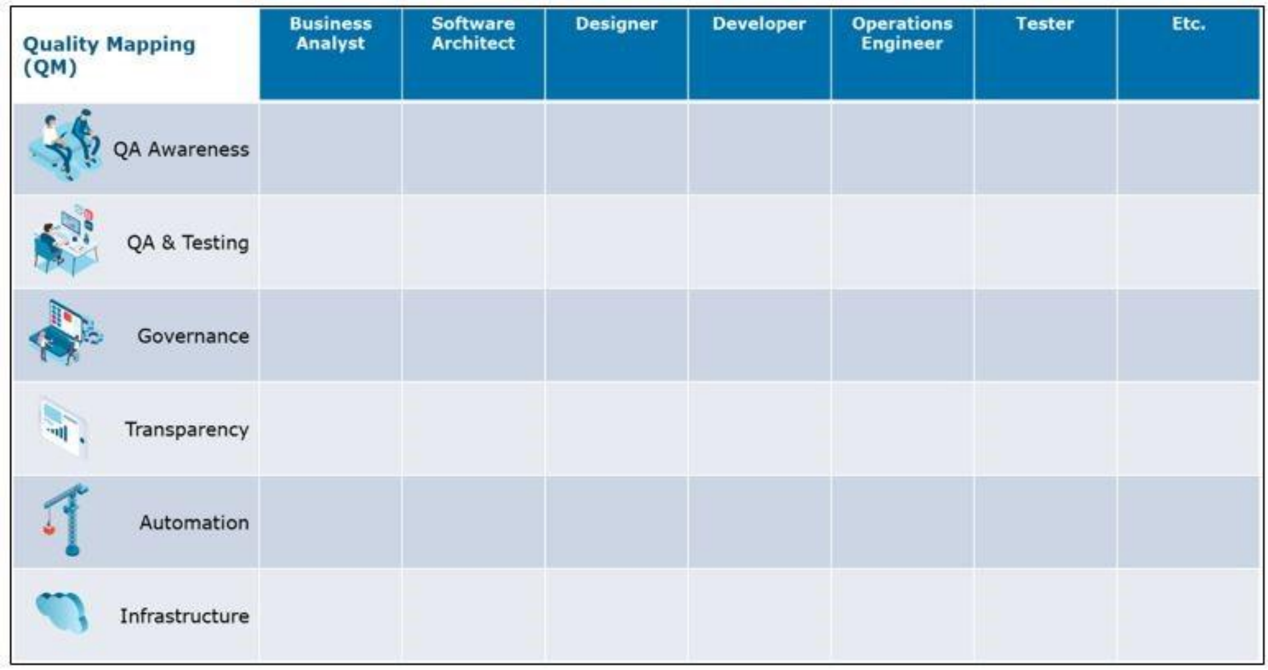
QM is a team effort and demands participation of all the people involved.

So, gather all people involved, which would be easy if you are Agile and work in a communicative and collaborative manner. You will see people with roles like business analyst, software architect, designer, developer, operations engineer, tester, database administrator and so on in the group. (*"Hey, aren't these people always there, no matter which framework or software development approach is used?"*)
Let every participant fill in — in the QM table — how they would add substance to the Quality key areas, and request a quality architect to be the moderator of the QM workshop as not all participants will be Quality experts.

How does a QM table look like?

Figure 1 shows a QM table that maps the quality key areas with various roles. At the intersections, the participants will mention the quality measures that they will apply. The end result is a table that you can use to provide clarity (transparency) to everyone about how each participant will contribute to the quality of software development. According to this table, the team can hold a certain person responsible if the promised contribution is not made.

I have used this approach several times, perhaps not always as successful, but it has always made people think about quality, which is of utmost importance to me. After all, it is people who create valuable high-quality software and not a framework.



Quality Mapping (QM)	Business Analyst	Software Architect	Designer	Developer	Operations Engineer	Tester	Etc.
QA Awareness							
QA & Testing							
Governance							
Transparency							
Automation							
Infrastructure							

Figure 1: QM Table Format

For now, I just wanted to explain the QM approach. Which quality measures can be chosen, is for another time. But you can think of: INVEST, TDD, SbE, ATDD, BDD, pair programming, test design techniques, risk analysis, risk poker, MBTD, MBT, code quality and test automation tools, review techniques, DoD, DoR, DoS, WoW, and so on.

Given the simplicity of QM, I encourage you to apply this — or experiment with it — immediately. I am sure you will be able to implement it even without further explanation.

Remember: *Don't let 'non-software development' frameworks distract you from the real goals of designing, building, running and maintaining a valuable high-quality software, preferably done by people with an Agile mindset and possibly supported by QM to achieve these goals!*

---

[1] The six quality key areas are borrowed from Sogeti's Agile Quality Improvement framework.

About Leo van der Aalst

I started in IT in 1987. I have successfully completed almost all roles from programmer to project manager and from innovation manager to department manager at a large number of (inter) national organizations. All this in waterfall, hybrid and agile environments. As QA&Test department manager I managed more than a hundred people. As an agile manager I advise organizations on IT, quality and test issues. Since 2005 especially in environments such as agile, scrum, DevOps, SAFe and Spotify. As an agile coach I provide training and workshops to more than 350 agile teams in the Netherlands and abroad and as a change manager I guide organizations in the transition to working agile. In addition, I am a much sought-after speaker and trainer (more than 1,500 workshops and more than 12,000 students in the last ten years), I am the author of five books on software quality and I develop and implement – agile – QA&Test improvement models.

More on Leo van der Aalst.

View my profile on Linked in