Received: 16 March 2016 | Revised: 23 May 2016 | Accepted: 1 September 2016

DOI 10.1002/smr.1823

WILEY Journal of Software: Evolution and Process

SPECIAL ISSUE – EUROSPI 2015

Effectivity and economical aspects for agile quality assurance in large enterprises

Alexander Poth

Volkswagen AG, Berliner Ring 2, D-38436 Wolfsburg, Germany

Correspondence
Alexander Poth, Volkswagen AG, Berliner Ring 2, D-38436 Wolfsburg, Germany.
Email: alexander.poth@volkswagen.de

Abstract

The transition from a classical to an agile software development procedure needs a structured and strategic roll out to realize the expected benefits of the transition phase. Quality assurance as part of the software development has also to be designed to realize its targets of the agile transition for adequate project and program quality assurance. Besides effectivity the economic aspects have to be implemented also in a value-driven agile product quality assurance. This tension between effectivity and economy will be shown on the example of the Volkswagen group IT. The transition is mapped to the SPI Manifesto to demonstrate the currency in 2009 of the established values and principles for software process improvement.

KEYWORDS

agile software development, agile transition, quality assurance, quality management, quality risk mitigation

1 | INTRODUCTION TO THE SPI MANIFESTO

The SPI Manifesto[4] has been written by a group of attendees of the euroSPI conference in 2009. The group identifies values and principles for software process improvement initiatives in companies.

The 3 values are as follows:

1. We truly believe that SPI must involve people actively and affect their daily activities.
2. We truly believe that SPI is what you do to make business successful.
3. We truly believe that SPI is inherently linked with change.

On the basis of these values the principles of Figure 1 are derived.

The importance of SPI with its values and principles is shown for example in the works of Garre-Rubio et al,[5] Hernández-López et al,[7] and Korsaa et al.[8]

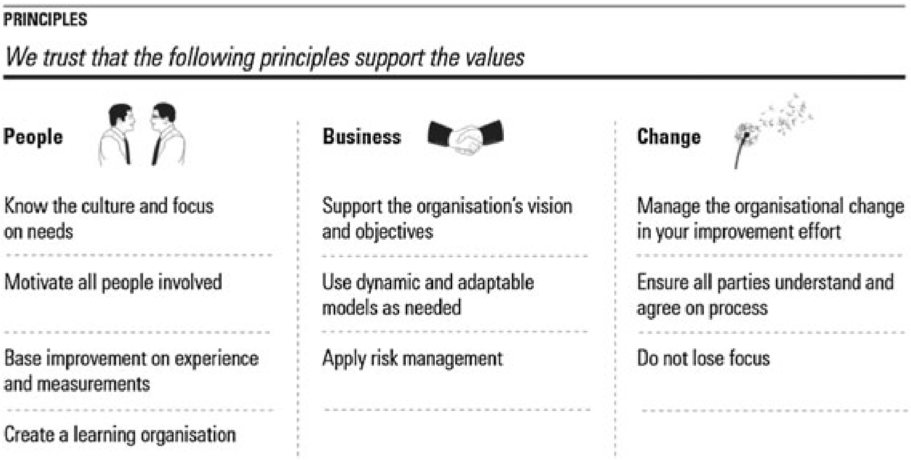
These principles are referred in the next chapters to show how an instantiation of the SPI Manifesto can be realized in a large enterprise organization context.

2 | AGILITY AT VOLKSWAGEN IT

The Volkswagen group IT is structured into the brands IT organizations with their own chief information officers. This article focuses on the IT of the brand Volkswagen and the IT at the group level. Further, this agile initiative is not focused on the electric/electronic development with their individual quality demands and regulatory for embedded software development.

The agile initiative is on the basis of a historically bottom-up–driven initiative, which was officially founded in 2012. The founding was formally realized with the establishment of an agile community (AC). This community meets approximately 10 times each year. The AC is under patronage of the senior management. It is contact point for new and established IT projects and programs. The community currently (2015) has more than 300 followers and more than 50 agile teams running IT projects and typically more than 80 participants in the community meetings. Everybody at Volkswagen AG is free to participate to the AC and their community of practices (CoP). A CoP is set up as special knowledge sharing group.

To set up an agile project or program the “readiness check” has to be fulfilled. This avoids the setup of an agile transition in a project or program environment, which is not capable to work in an agile fashion. For example, sometimes projects have an issue to identify 1 person as



PRINCIPLES

We trust that the following principles support the values

People	Business	Change
Know the culture and focus on needs	Support the organisation's vision and objectives	Manage the organisational change in your improvement effort
Motivate all people involved	Use dynamic and adaptable models as needed	Ensure all parties understand and agree on process
Base improvement on experience and measurements	Apply risk management	Do not lose focus
Create a learning organisation		

FIGURE 1 Principles of the SPI Manifesto from [eur09]

product owner. In this case it is not recommended to start a transition before the prerequisites are fulfilled.

The described context of the AC is our base to fulfill the principle "know the culture and focus on needs."

The agile center of excellence (ACE) is our facilitator for all organizational issues. It is on the basis of representatives of each IT department. The ACE developed with the AC the "IT Produktentstehungs-Prozess (PEP) agile" (see Figure 2) as agile development process for the Volkswagen group IT. The IT PEP agile is also maintained by the ACE and the AC who adjusts and improves it. Further, the ACE has a pool of experienced agile coaches who are helping the projects and programs during transition phase to establish a working agile environment for their product development. The transition to agile development sets up only 3 roles[11]:

1. Product owner
2. Scrum master
3. Development team

These 3 roles build a self-organized team that cares about the agile values and the product. The quality assurance (QA) role and other dedicated classical roles are part of the development team. This reduces the high amount of specific domain roles with T-shaped[9] profile of the scrum team members.

Due to the team setup and their opportunity to (co-)create, their development environment via the AC and the CoPs is addressed the principle "motivate all people involved."

To establish and build knowledge for the agile context the ACE offers different training programs for scrum masters, product owner, team members, and management 3.0.

The AC, CoPs, and ACE with the coaches and the training programs are the platform for inherent quality.

With the ACE, AC, and agile coach it is addressed the principle, "manage the organizational change in your improvement effort."

3 | GUIDE THE TRANSITION

The ACE agile coach addresses the principle "base improvement on experience and measurements." The coaches are a service, which is offered by the ACE to the projects for the transition phase. Depending on the project or program size and their starting position on the basis of the readiness check, the projects can select from small to large coach packages (see Figure 3). The different packages are for adequate support of the project transition. The coaches are experienced in different domains and project sizes to transfer know how to the agile teams which are setup. The coach is able to terminate a transition in case of impediments, which are show stoppers for the agile transition. If the transition is successful the projects get an "ACE approved." "ACE approved" is used to separate projects that are really agile according to the agile values from many pseudo-agile projects that want to use the word agile to work without in the worst case without any values, standards, and rules, which easily leads to chaos. The success rate of pseudo-agile projects is not as high as real agile projects, and the

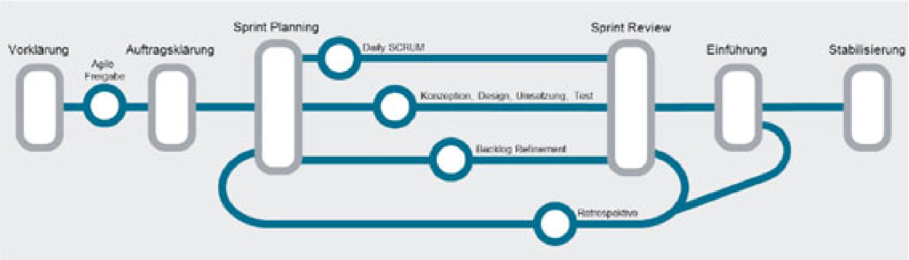


FIGURE 2 Extract of the IT PEP agile handout in subway-schema (German words are informative)

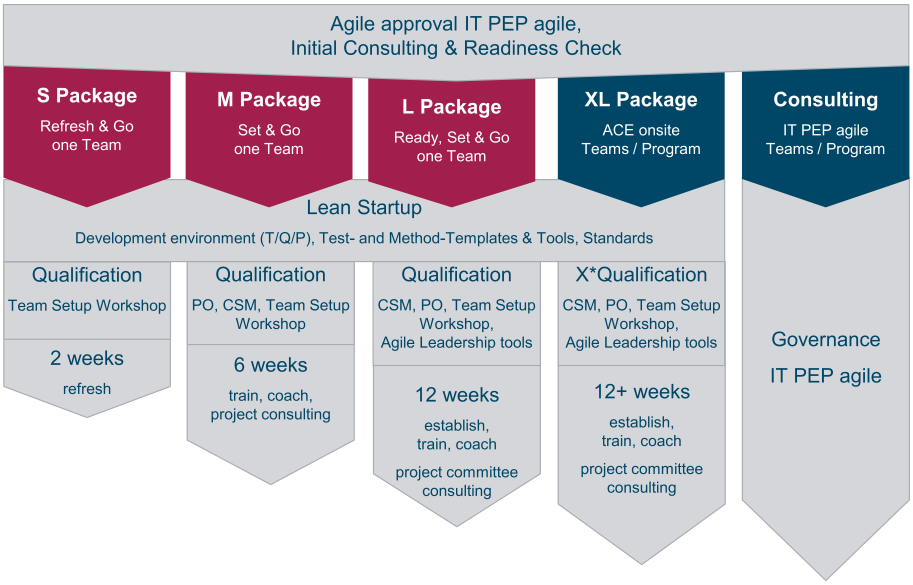

Agile approval IT PEP agile,
Initial Consulting & Readiness Check
S Package
Refresh & Go
one Team
M Package
Set & Go
one Team
L Package
Ready, Set & Go
one Team
XL Package
ACE onsite
Teams / Program
Consulting
IT PEP agile
Teams / Program
Lean Startup
Development environment (T/Q/P), Test- and Method-Templates & Tools, Standards
Qualification
Team Setup Workshop
2 weeks
refresh
Qualification
PO, CSM, Team Setup Workshop
6 weeks
train, coach, project consulting
Qualification
CSM, PO, Team Setup Workshop, Agile Leadership tools
12 weeks
establish, train, coach
project committee consulting
X*Qualification
CSM, PO, Team Setup Workshop, Agile Leadership tools
12+ weeks
establish, train, coach
project committee consulting
Governance
IT PEP agile

FIGURE 3 Coach packages for project transition phase support

"ACE approved" saves the word agile against this high risk to fail projects.

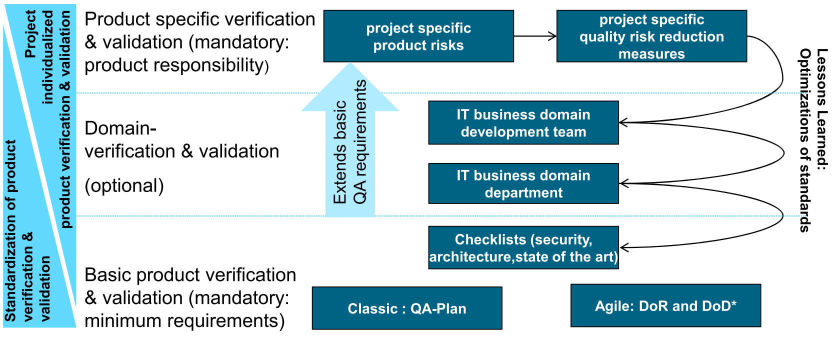
To coach a team to agile development from the QA perspective, the ACE offers some methods to handle in an agile fashion QA aspect and artefacts. To address the estimated established QA artefacts, a mapping is a first step to identify what outcomes are needed to fulfill general enterprise processes. These mapping shows, for example, that the content of the QA plan maps mostly with definition of ready (DoR) and the definition of done (DoD), or a test plan in an agile fashion could be a systematic management (identification and mitigation) of product quality risks. Do not forget the agile mind set: small iterations (sprints) have no huge changes and do not need heavy QA activities.

The SPI principle, "ensure all parties understand and agree on process," is handled with the methods and tools, which the ACE has in place. With the IT PEP agile a generic development reference model shows the agile "how to." With tools like HP ALM with an agile project template and Atlassian Jira, a direct support for the IT PEP agile methods is given to the projects. Further there are best practices in the intranet available. The agile coach coaching the projects in the transition phase to support the mind change of the team members and support the new orientation to IT PEP agile etc further the coaches help the projects to adapt these generic methods and tool templates. On the other hand the coaches get ideas to improve the methods and tools for more easy use and adaption.

3.1 | AGILE QUALITY MANAGEMENT

The SPI principle, "apply risk management," is used on both levels: during the transition process and in the final development process. This is addressed by the objective that product verification is the reduction of quality risks and the evidences for required product features. This leads to the mindset that quality responsibility comes from product quality. With this mindset the ACE establishes a basic set of standards, and the domains and projects extend this (Figure 4). Via feedbacks and lessons learned, the domain and basic set could be adapted.

On the basis of this QA environment the scrum master is responsible for an efficient product realization. His job is to initiate changes for more efficiency and also for QA aspects. With this focus QA activities are part of the product backlog-like load, and performance testing, test automation, and product quality risk mitigation activities are planned via tasks on the scrum board. After a view sprints it is often visible that QA tasks–like performance benchmarking are things for every sprint and release. Routine QA


Project individualized product verification & validation
Standardization of product verification & validation
Product specific verification & validation (mandatory: product responsibility)
Domain-verification & validation (optional)
Basic product verification & validation (mandatory: minimum requirements)
project specific product risks
project specific quality risk reduction measures
Extends basic QA requirements
IT business domain development team
IT business domain department
Checklists (security, architecture,state of the art)
Classic : QA-Plan
Agile: DoR and DoD*
Lessons Learned: Optimizations of standards

*Definition of Ready or Done

FIGURE 4 Relation of product individual and generic quality activities

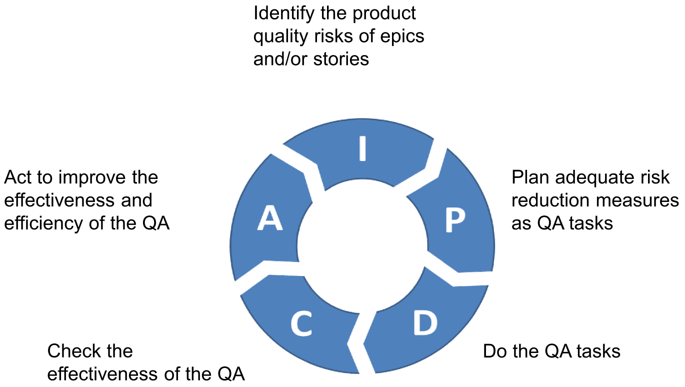

Identify the product quality risks of epics and/or stories
I
Plan adequate risk reduction measures as QA tasks
P
Do the QA tasks
D
Check the effectiveness of the QA
C
Act to improve the effectiveness and efficiency of the QA
A

FIGURE 5 Schematic product specific quality risk identification, mitigation, and improvement

activities also can be planned with Kanban[1] instead of scrum. To realize product specific value drivers the QA framework is open in methods and technologies as scaling and efficiency drivers for QA. This leads, for example, to an intensive support for the continuous integration platform by quality stakeholders to improve the (test) automation and fast feedback cycle about builds. On the basis of this QA environment the SPI principle, “use dynamic and adaptable models as needed,” is applied during the QA transition and the project and product life cycle.

The SPI principle, “create a learning organization,” is realized on the product level with the extended plan-do-check-act cycle (Figure 5). The classical 4-phase plan-do-check-act cycle[3] is extended by the identification phase for systematical product quality risk identification and analysis. With the product quality risks the product quality is focused with their product specific business domain and technical quality risks. Each function/feature is evaluated to the product quality risks, and adequate validation measures are defined to reduce or mitigate the risk. Further the result is monitored with the product quality indicators like defects in production. Due to the agile iterative product development this IPDCA cycle[10] sets up a learning organization about the product quality. The typical agile team retrospective focuses on the improvement of the development team organization.

To mitigate the product quality risk in an efficient way the validation measures are well-known standard QA methods like reviews,[6] unit testing,[12] and product domain specific QA methods and tools like foot printing of server/client communication calls. Other product domains need special wide area network simulations for performance and load testing to assure positive plant roll outs. With this foot-printing and run-time logging release and build specific benchmarks are made. With the openness of the QA transition the ACE addresses the SPI principle, “support the organization's vision and business objectives.”

4 | EFFICIENCY IMPROVEMENTS WITH AGILE QA METHODS

The idea behind the agile QA is to integrate QA activities into the agile development team tasks to reduce double work and waste. This inherent QA leads to high transparency of done QA tasks and with the small product increments (sprints) and their reduced complexity in the increments, it is possible to reduce hand-over capacities of dedicated test phases. This transparency makes it possible to test many functional and integration aspects in the development context via, for example, j-unit tests (Figure 6 blue parts). In cooperation to the classic formal big integration and user acceptance tests, the agile teams focus only on the missing parts and can reduce this kind of tests (green parts), and this optimization reduces the overall QA efforts.

This leads to more efficiency for the aspects of time to market and costs. The following are methods and aspects that are supporting more efficiency:

1. Test case derivation
   - **a.** Risk-based test focus derived with identified domain and technical quality risk.
   - **b.** Reuse focus is realized with test scenarios, which are based of reusable test case modules.
2. Measures to build quality on development artefacts
   - **a.** Static code analysis for each build
   - **b.** Unit testing for functional and integrative tests
   - **c.** Non-functional–requirement testing for example for security, load, and performance aspects
3. Test organization
   - **a.** Supplier transparence about quality measures for “retest optimization”

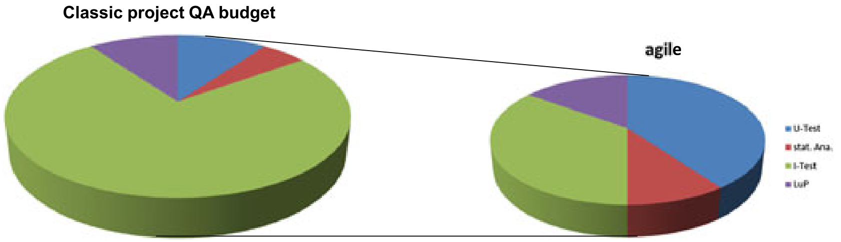


FIGURE 6 Schematic view on quality and business efficiency

   b. Integration of test analysts into feature teams for adequate test case development
   c. Constant interaction of the quality management with the development team and the suppliers (active supplier management for quality aspects)
4. Technical debt analysis
   a. Algorithm for technical debt reduction: commit first the fixing of return on investment technical debts then “haggling” about others (return on investment for fixing the debt is calculated against do nothing costs of living with the debt).

5 | CONCLUSION AND SOME ASPECTS ABOUT KEY PERFORMANCE INDICATORS

The QA transition addresses the SPI principle, “do not lose focus,” with its vision about effective and efficient QA. The vision about the QA focus is shown in Figure 6 with the comparison of QA efforts in staged classic projects and iterative and incremental agile projects. To reach this objective everything is aligned during the transition and later in the project development to identify improvement potentials. But the ACE keeps in mind DeMarco's observation[2] that if a quantitative target is defined—for example, via management by objectives—there is typically a security buffer, and you will not get what is possible and mostly not more as defined in the target in real project life.

To observe that the transition and the projects are on the right way about the QA objective, different indicators are tracked beyond established agile metrics like team velocity. Depending on the project these indicators are tracked in a more or less formal way:

1. Readiness check of ACE
   a. Must be positive as starting point for an agile transition
2. Code quality (IT development internal loop)
   a. #broken_rules of static code analysis over sprints and releases (leads over time to a burn down chart also if legacy code is measured because of the refactoring which fixes some historical issues)
3. Known errors from the operation team (external customer loop)
   a. #defects sorted by severity over time (some projects have zero as target)
   b. Root cause to the component (identify systematic errors and hot spots)
4. Technical debts
   a. #debts sorted by severity over time is a burn down chart

As a restriction for all aspects of an agile transition, the ACE identifies the culture of the organization. You cannot be agile if the environment and management think and act in classical patterns and fashion.

To summarize the tour about the transition of QA from classical to agile project development, this is a good way to refocus the QA itself and its over years–developed QA methods and tools to a lean and value oriented QA, which leads to a more product centric QA. Further it is an exercise about SPI for the QA organization to reflect one of the quality topics, which are often given as an advice to “other guys” with quality issues.

REFERENCES

1. Ahmad O, Markkula J, Oivo M. Kanban in software development: a systematic literature review. *IEEE Proc of SEAA*. 2013;
2. DeMarco T. Software engineering: an idea whose time has come and gone? *IEEE Softw*. 2009;26(4):95–96.
3. Deming WE. *Out of the Crisis. Massachusetts Institute of Technology*. Cambridge: Cambridge University;1982.
4. euroSPI et al. SPI Manifesto. euroSPI.net, 2009.
5. Garre-Rubio M, García-Barriocanal E, Siakas K, et al. Analysing the corporate responsibility web pages of consumer electronics companies: implications for process improvement. *Software, IET*. 2012;6(5):451–460.
6. Gilb T, Graham D. Software Inspection. Addison Wesley, 1994.
7. Hernández-López A, Colomo-Palacios R, Soto-Acosta P, Casado-Lumbreras C. Productivity measurement in software engineering: a study of the inputs and the outputs. *Int J Information Techn Sys Approach*. 2015;8(1):45–67.
8. Korsaa M, Johansen J, Schweigert T, et al. The people aspects in modern process improvement management approaches. *J Softw Evol Process*. 2013;25(4):381–391.
9. Madhavan R, Grover R. From embedded knowledge to embodied knowledge: new product development as knowledge management. *J Mark*. 1998;62:1–12.
10. Poth A, Sunyeav A. Effective quality management: value- and risk-based software quality management. *IEEE Softw*. 2014;31(6):79–85.
11. Schwaber K, Beedle M. *Agile Software Development with Scrum*. Prentice Hall;2002. ISBN 978-0-13-067634-4
12. Vegas S, Juristo N, Basili VR. Maturing software engineering knowledge through classifications: a case study on unit testing techniques. *IEEE Trans Softw Eng*. 2009;35(4):551–565.

How to cite this article: Poth, A. (2016), Effectivity and economical aspects for agile quality assurance in large enterprises, *J Softw Evol Proc*, 28:1000–1004. doi: 10.1002/smr.1823