



Clarifying Myths with Process Maturity Models vs. Agile

White Paper

Version: **1.0**

Release date: **Aug 6th 2014**

Distribution: **Public**

Status: **Released**

Authors (alphabetical order): **Frank Besemer, SynSpace GmbH**
Timo Karasch, Method Park Consulting GmbH
Pierre Metz, Brose Fahrzeugteile GmbH & Co. KG
Joachim Pfeffer, SynSpace GmbH

Content

- 1 Motivation.....3
- 2 Brief Intro to “Process Maturity Models”3
- 3 Brief Intro to “Agile Development”4
 - 3.1 Agile Development4
 - 3.2 Agile Approaches.....4
- 4 Levels of Abstraction of the Term “Process”5
- 5 Resolving Myths6
 - 5.1 “SPICE Requires a Waterfall Model”6
 - 5.2 “SPICE Requires the Creation of Extensive Documentation”6
 - 5.3 “SPICE Requires Rigid Process Standards”7
 - 5.4 “Agile Contradicts SPICE and Vice Versa”7
 - 5.5 “In Scrum (and XP & Co) there are no Defined Processes”8
 - 5.6 “When Working Agile, Documentation Can Be Neglected/Omitted”8
 - 5.7 “Agile / SCRUM Does Not Work In Large or Multi-Sited Projects, Therefore SPICE Is Needed”9
 - 5.8 “Agile Practices Empower Individuals While SPICE Impairs People by Making Them Exchangeable”9
 - 5.9 “The ‘Command-And-Control’ of SPICE Is Incompatible With Agile Development” 10
 - 5.10 “In Scrum ‘Continuous Improvement’ is Part of the Framework and Therefore Scrum Reaches At Minimum SPICE Level 3 or Even Level 5” 10
- 6 Conclusion..... 10
- 7 About the Authors..... 11
- 8 Appendix A – Understanding Process Capability Levels (CL) 11
- 9 References..... 13

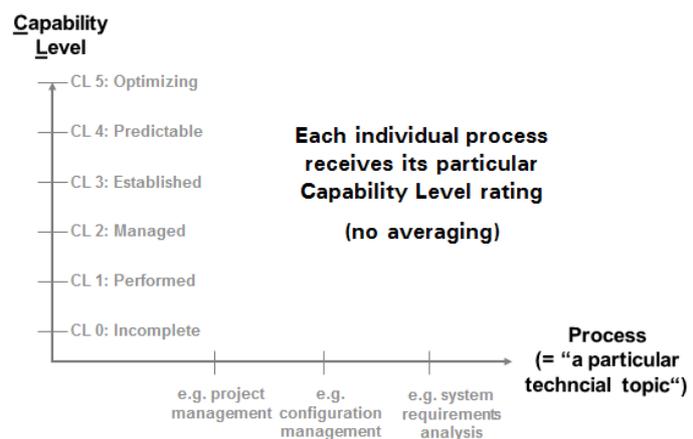
1 Motivation

For more than one decade there has been a lot of controversy in respect to Agile practices and process assessment / process maturity models such as CMMI and ISO/IEC 15504-2-compliant models such as ISO/IEC 15504-5, -6, or Automotive SPICE®. The reasons may be various, e.g. mutual half-knowledge of protagonists, lack of sufficient professional experience, or even the intention to reach “political” goals in organizations etc. Unfortunately, this hampers embracing and capitalizing on the process and method state-of-the-art, sometimes even leading to recurring and prevailing “fundamentalism” or “religious wars”.

For these reasons, in section 5 of this white paper we try to clarify the most frequent “myths” we have encountered, highlight their root causes and offer answers from our perspective based on a discussion of the term “process” at certain levels of abstraction. As an important prerequisite, in sections 2 and 3 we orientate the reader about SPICE and Agile.

2 Brief Intro to “Process Maturity Models”

People work in companies that want to release products to the market in order to earn money. Today in distributed development environments and in a highly specialized world companies make use of development suppliers. The problem: companies need criteria for selecting suppliers. Criteria involve technical experience, economic aspects, strategic goals, process maturity and many more. Process maturity is one out of many relevant criteria because it contributes to product quality: the more structured one works the lower the likelihood is of introducing systematic faults into the product and the more accurate economically-relevant estimations are. Therefore, since the 1980s successful industry projects were compared worldwide, and their success factors and good practices were identified. Process maturity models such as CMMI (originated in the USA), or ISO/IEC 15504 SPICE (European initiative) and its industry sector-specific derivatives have organized those factors and good practices in terms of abstract interconnected principles/goals in two dimensions, see Figure 1.



Processes are defined in terms of a purpose and corresponding outcomes. Examples in Automotive SPICE® v2.5 are:

- Project Management
 - *Define project life cycle: ... which is appropriate to the scope, context, magnitude and complexity of the project. [Outcome 2]*
 - *Review and report progress of the project: regularly report and review the status of the project against the project plans to all affected parties. [Outcome 6]*
- Software Construction
 - *Define a unit verification strategy. ... for verification and re-verifying the software units. The strategy should define how to achieve the desired quality with the available and suitable techniques ... [Outcome 1]*

Each of these “processes” can be performed at 6 different levels of excellence, called Capability Levels, (see Section 8 for more details). CL 1 requires the achieving of the process-specific outcomes. CL 2 and higher are generic, examples e.g. at Capability Level 2¹ are:

- *Define the requirements for the work products: ... to be produced are defined. Requirements may include defining contents and structure. Quality criteria of the work products are identified. Appropriate review and approval criteria for the work products are defined.*
- *Define the requirements for documentation and control of the work products: ... Such requirements may include “storage, access rights, requirements for distribution, versioning & baselining, work product status & validity periods, archiving, traceability” [6] etc.*

Consequently a process maturity model is a mere tool for measuring process maturity against internationally reconciled abstract principles. The results may be used for benchmarking, supplier selection and, also, for providing orientation for investing in process improvements within a company.

3 Brief Intro to “Agile Development”

3.1 Agile Development

Agile development is a group of development practices based on iterative and incremental development, where requirements and solutions evolve through collaboration between self-organizing, cross-functional teams. It promotes adaptive planning, evolutionary development and delivery, a time-boxed² iterative approach, and encourages a rapid and flexible response to change as well as fast feedback loops for the customer and the development team.

3.2 Agile Approaches

The most popular Agile development approaches include:

Scrum

Scrum is a framework for product development with tangible artifacts like Product Backlog, Sprint Backlog, Product Increment and Definition of Done [3]. Scrum focuses on the iterative incremental delivery of value. Several meetings (Events) are conducted repeatedly for each Sprint (i.e. a 1-4 week time-box to build a new product increment): the Sprint starts with the Sprint Planning meeting. At the end of the Sprint the Sprint Review meeting is conducted where the newly created Product Increment will be evaluated together with the customer and feedback is gathered. It is followed by the Sprint Retrospective where the whole Scrum Team (consisting of the Product Owner, a Scrum Master and the Development Team) reviews the processes and improves them. During the Sprint a daily meeting is conducted where the Development Team synchronizes its work and makes the planning for the next day.

Kanban for Development

Kanban [4] is an emerging set of practices for development teams derived from Lean Manufacturing, the Toyota Production System (TPS), and Goldratt’s Theory of Constraints. Kanban uses visual controls

¹ Aspects like a company’s business goals come into play at Capability Level 4. See section 8.

² Time-boxing is a project planning technique applied when a deadline is fixed and therefore the resources and goals are the only variables. The schedule is divided into a number of separate time periods (timeboxes), with each part having its own deliverables, deadline and budget.

(either by paper cards on a board or a Kanban software tool) to illustrate and control the flow of work and its bottlenecks in order to remove them. It promotes to “limit your work in process” in order to improve the throughput/cycle time. It focuses on the continuous delivery of value.

eXtreme Programming (XP)

XP aims at enabling successful software development despite vague or constantly changing software requirements. Some of the main characteristics of XP are short iterations (i.e. one week) with small releases and rapid feedback, close customer participation, constant communication and coordination, continuous refactoring, continuous integration and testing, collective code ownership, and pair programming.

4 Levels of Abstraction of the Term “Process”

When looking at sections 2 and 3 we realize that it is hard to spontaneously and intuitively differentiate between the terms “process”, “method”, and “approach” thus making it difficult to understand the differences between Process Maturity Models and Agile.

Therefore, in Figure 2 we want to offer three levels of abstraction for the term “process” [1] that we find helpful to explain reasons of, and offer clarifications for, myths that are frequently encountered. Please note that these levels of abstraction are not meant to define a strict black-or-white demarcation line, nor is it the aim to provide a scientific classification scheme – the message here is to understand that, in practice, when it comes to the term “process” there are different abstraction levels.

Writing down experiences made during product development (i.e. at the DOING level) in order to share these experiences with others means creating a HOW level.

However, a HOW is always specific to a particular context such as a company or a product line. For example, the HOW of company A is certainly not applicable as is to company B. However, both companies

can be expected to e.g. maintain work product versioning and product documentation baselines against which e.g. change requests can then be placed. These principles are at the WHAT level while deciding on solutions for concrete templates, proceedings, and tooling etc. is left to the HOW level.

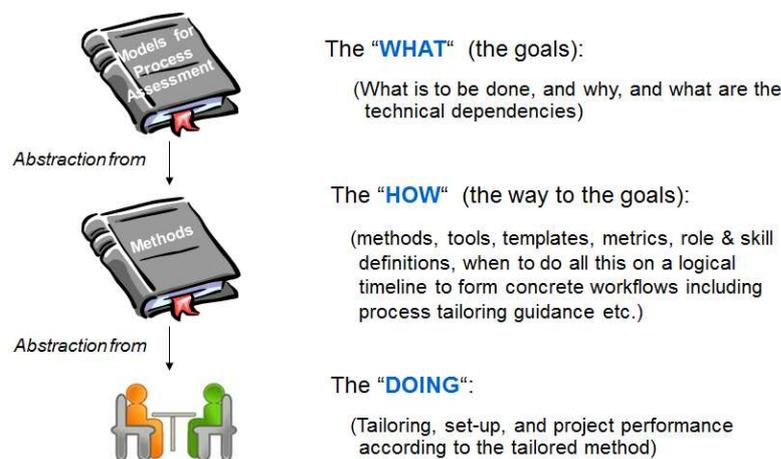


Figure 2: Understanding of the term “process” at three levels of abstraction [1]

When comparing sections 2 and 3 we learn that Process Maturity Models reside at the WHAT level while Agile approaches rather are at the HOW level. This should help to clarify the myths as we will see in the next section.

5 Resolving Myths

5.1 “SPICE Requires a Waterfall Model”

Myth

SPICE is said to require a waterfall model lifecycle while iterations (reworking or refining existing product functionality/system functionalities) and incremental product development (stepwise adding new, or completely removing, product functionality/system functionalities) are not allowed or possible; further it is said that SPICE requires specific work products to be available at predefined points in time.

Our Perspective

Since SPICE is at the WHAT level it cannot, and does not, predefine any lifecycle model. Consequently, it does not predefine any logical points in time at which work products shall be available, nor does it predefine any other kind of activity sequences or ordering. Instead, SPICE requires the selection and usage, of any reasonably appropriate lifecycle model defining such sequences (see examples on project management in Section 2). The actual, and meaningful, selection is a decision at the HOW level made by the project or the organization.

Further, SPICE offers principles at the WHAT level that allow organizing changes of any kind to work products and products (SUP.10) which supports iterative and incremental development at the HOW and DOING levels.

5.2 “SPICE Requires the Creation of Extensive Documentation”

Myth

SPICE lists so called “work products” including “work product characteristics” (work product content expectations). This means all these work products/documents need to be created in the same way and according to that very structure, as otherwise SPICE would not actually include such lists. Further, since the work product characteristics are comprehensive it is obvious that overly extensive documentation is the result.

Our Perspective

Firstly, at the DOING level documentation is generally required in order to e.g. conserve the project history and product knowledge to have a basis for discussing change requests with customers, and to support the organization in providing evidence in the context of e.g. product liability cases, or to satisfy homologation requirements. In this respect, there is no difference between employing SPICE or Agile.

Secondly, with respect to the work product characteristics themselves, SPICE aims to

- support and assist the assessors in identifying the presence or absence of principles (as depicted at the WHAT level) in the assessed HOW and DOING.
- reduce assessment results variance (this can be caused by either an assessor being so experienced in a particular topic so that he “imposes” his detailed technical expectations on the assesses thereby inadvertently leaving the boundary of the WHAT level; this may also be caused by the fact that an assessors mostly is not at the same time an expert in design/testing/programming/project management/quality assurance etc.).

In order to do that SPICE offers the assessor two types of “indicators”, namely “Base/Generic Practices” (being activity-oriented indicators) and “work products & characteristics” (being result-oriented indicators).

These very comprehensive work product characteristics (being the reason of the myth) are meant to offer a good practice and state-of-the-art knowledge set at the WHAT level for assessor guidance; therefore, they are organized in “some” fashion that is believed to be a quickly accessible information source during an assessment. In that respect they represent examples only. Therefore, work product & characteristics are neither a “strict must” nor are they normative for organizations. Instead, the actual structure, form and content of work products and documents must be decided by the organization ensuring that it is appropriate for the intended purpose and needs of its projects and product development goals. This means that documentation that is perceived as “extensive” or “without purpose” is more a sign of not being pragmatic and of missing added value.

It is the assessors’ sole responsibility to map the HOW and DOING levels to the SPICE indicators.

Thereby, in fact, a good SPICE assessor is even able to, and will, highlight over-engineered processes and documentation.

5.3 “SPICE Requires Rigid Process Standards”

Myth

SPICE predefines precisely how product development shall be performed in terms of there being only one allowed “proceeding” the organization/the projects have to adhere to – changes or adaptations are prohibited. As a result, SPICE impedes innovation and creativity and, thus, hampers advancements in process technologies, methods, and efficiency.

Our Perspective

Since SPICE is at the WHAT level it cannot, and does not, predefine any concrete workflows or sequences of activities. Concrete workflows etc. are at the HOW level. In that respect, the WHAT level generally allows an “infinite” number of proceedings to exist.

If, however, an organization decides for their HOW level to agree on “standards” in order to exploit the advantages coming with it (see Section 8 on Capability Level 3), SPICE still allows various standards to exist for particular scenarios such as different customers, low/medium/complex projects, commodity products vs. innovative products, series development vs. explorative prototype development, etc. Further, when one of those standards is chosen by a project SPICE further promotes the (reasonable and meaningful) tailoring of that standard to the individual needs of the project’s context (this is because HOW-level standards still are more generic than the reality of a DOING as otherwise standards would not be applicable across projects).

Furthermore, for any HOW-level standard SPICE requires an actually used feedback cycle to be in place for collecting the experiences from the process performers in order to continuously and critically reflect on standard adequacy, and adapt or advance the standard, if necessary (process improvement).

5.4 “Agile Contradicts SPICE and Vice Versa”

Myth

Agile and SPICE are not compatible and inconsistent with each other. Therefore, a combination of both in practice is impossible.

Our Perspective

Since SPICE resides at the WHAT level while Agile practices are mainly at the HOW level both domains cannot, by definition, contradict each other. In fact, Agile practices “implement” some of the SPICE principles, and SPICE principles “serve as an abstraction” of Agile elements.

This myth appears to be simply caused by mutual misinterpretations and half-knowledge of individuals in both domains.

5.5 “In Scrum (and XP & Co) There Are No Defined Processes”

Myth

Team members may change the way they work from day to day. No rules exist. There are no mandatory process elements or documentation.

Our Perspective

First of all, as explained in chapter 3.2, Scrum and XP do suggest certain roles, artifacts and activities which actually make up “rules of the method”. These are elements of any process definition. Employing XP or Scrum therefore means to implement, and follow, these elements as otherwise one would not actually employ this method.

Secondly, adapting these methods in terms of adding additional activities, artifacts, roles & responsibilities etc. (“adding rules”) is recommended and is common practice. In the context of professional product development any adaptation is deliberately made whenever it is considered to be meaningful, necessary, and of added value which is not arbitrariness.

5.6 “When Working Agile, Documentation Can Be Neglected/Omitted”

Myth

One of the statements in the Manifesto for Agile Software Development [2] is that “working software is valued more than comprehensive documentation”. This means that creating documentation beyond the software product, e.g. specifications, design, verification/validation reports in digital or physical form, is not a goal.

Our Perspective

Firstly, this myth is apparently based on a misinterpretation of the quoted statement. In fact, it neither states nor means that documentation may be neglected. It simply means that working software and comprehensive documentation are two co-existing values which should be balanced in terms of pragmatism, added-value, and organizational interests such as being able to show evidence e.g. in product liability cases. This has nothing to do with omitting information.

Further, as explained in chapter 3.2, Scrum and XP do suggest particular artifacts/work products with particular content and expected quality. Employing XP or Scrum therefore means to create tangible information as otherwise one would not actually employ this method. As we see, using an Agile practice while disregarding its required artifacts would be inconsistent.

5.7 “Agile / SCRUM Does Not Work In Large or Multi-Sited Projects, Therefore SPICE Is Needed”

Myth

Scrum and other Agile practices focus on single teams with less than 10 team members. Practices for the collaboration of several teams, or any other scaling concept, is not provided by Agile approaches. However traditional project management and SPICE processes are designed to handle large projects.

Our Perspective

We would like to provide two counter arguments:

Firstly: what do Agile approaches provide to scale up development? While pure Scrum and pure XP implement agility at the team level further concepts arose that allow the scaling up of agility to larger teams or even geographically distributed environments. Examples are “Scrum of Scrums” or the Scaled Agile Framework (SAFe).

Secondly: what does SPICE provide to scale up development? SPICE being a measurement tool of process maturity does not provide any concrete proceedings solution for any context because SPICE defines the WHAT level. Defining any method, proceeding, or workflows is an action that takes place at the HOW level.

5.8 “Agile Practices Empower Individuals While SPICE Impairs People by Making Them Exchangeable”

Myth

SPICE requires processes that allow easy exchange of project members in a “de-humanizing” manner while Agile approaches promotes to empower individuals.

Our Perspective

Firstly, Agile practices promote collaboration and collective achievements but not “hero engineering”. Secondly, SPICE expects *any* process performer to provide critique on the used process approaches in order to drive improvement.

In fact, there is no statement in Agile practices or in Process Maturity Models that either directly or indirectly supports this myth. Rather, this myth simply appears to be the result of mutual arbitrary and pessimistic, and maybe sometimes even hostile, interpretations of individuals in both domains. The arbitrariness with such interpretations becomes obvious when considering another arbitrary but *optimistic* interpretation such as “SPICE optimizes and simplifies work and therefore makes people happy, so does Agile”. However, SPICE resides at the WHAT level and thus is subject to meaningful and pragmatic implementation while Agile practices, being at the HOW level, are a contribution to such implementations. As always, it is up to the organization to make what it can out of these practices and suggestions.

5.9 “The ‘Command-And-Control’ Attitude of SPICE Is Incompatible With Agile Development”

Myth

Higher levels of management are sometimes said to employ SPICE in order to gain control over people or otherwise measure an individual’s performance. Based on this, it is often argued that developers try to enforce an Agile approach to “defend” themselves.

Our Perspective

There is no statement in SPICE that, neither directly nor indirectly, requires, supports or promotes a ‘command-and-control’ attitude; in fact, since SPICE is at the WHAT level it cannot, by definition, pre-define any concrete approach or concrete HOW. The subject of SPICE and ISO/IEC 15504 is, on the contrary, to enable process improvement, it does not address “measurement of people”. Using SPICE for “measurement of people” is a false understanding or even misuse.

Rather, this myth (see also 5.8 here) appears to be driven by the observation that sometimes the SPICE principles are intentionally, or accidentally, but certainly inadequately interpreted or implemented which is in conflict with the above mentioned philosophy of Process Maturity Models.

5.10 “In Scrum ‘Continuous Improvement’ is Part of the Framework and Therefore Scrum Reaches at Minimum SPICE Level 3 or Even Level 5”

Myth

Scrum and other Agile approaches foster, or even, demand continuous evaluation and improvement of the current process. SPICE demands such improvement cycles only at Capability Level 5 being the highest achievable level. Since Scrum strives for continuous improvement just as SPICE does Scrum enables to reach SPICE Capability level 5 with very little effort.

Our Perspective

Yes, Scrum provides concepts and procedures for continuous improvement. However, Scrum only targets at a subset of the SPICE level 2 requirements. For example, the following aspects of SPICE level 2 are not covered by Scrum:

- Change control for work products
- Versions of work products
- Transparent revision status
- Baselines

With pure Scrum (i.e. without any extensions) SPICE level 2 cannot be fully achieved which is however a prerequisite for being able to fully benefit from SPICE level 3. Therefore level 5 can’t be achieved either (see Section 8). However, Scrum may achieve higher capability levels when complemented with other practices or procedures.

6 Conclusion

We have offered resolutions of myths about Agile practices vs. Process Maturity Models revealing that their root causes are due to misunderstandings or mutual half-knowledge in both domains. All this may be human and therefore not completely avoidable – however, the models cannot be blamed for it. Further, these clarifications should lead us to the understanding that the general message is not to “just use

a model or practice as is” or “to be right”. We believe that the overall message is to support people and organizations to become better. Thus, we believe that it is necessary to wholeheartedly and without bias study and experience good practices and state-of-the-art in order to be able to deliberately select what is beneficial to the respective professional environments and individual project contexts one is facing. Contexts vary, so do “best solutions”. In this respect, this paper has shown that SPICE and Agile complement each other so practical solutions should combine and exploit both sources.

7 About the Authors

Pierre Metz is an intacs®-certified Principal Assessor, a trainer for the Provisional and Competent Assessors levels, and an instructor for trainer candidates for both levels. He also is a member of of the intacs® Advisory Board and the VDA Working Group 13. Furthermore, Pierre is the head of the intacs® Working Groups “Syllabi” and “Standard Training Materials”. pierre.metz@brose.com

Frank Besemer is an intacs®-certified Principal Assessor. He supports development teams in implementing Agile/Lean methods, not only at the team level but also at the organizational level. He is a member of the intacs® Working Group “Assessor Support” and the head of the special interest group “Agility” in the federal state Baden-Württemberg, Germany, of the ASQF (“Association for Software Quality and Further Education). Frank.Besemer@synspace.com

Joachim Pfeffer is an intacs®-certified Provisional Assessor and a certified Professional Scrum Master. At SynSpace Joachim helps customers to implement Agile practices like Scrum, Kanban, Lean, and Theory of Constraints in compliance-driven development areas. Joachim is a visiting lecturer at the University of Weingarten-Ravensburg. Joachim.Pfeffer@synspace.com

Timo Karasch is an intacs®-certified Competent Assessor for Automotive SPICE®. His challenge is to combine topics like SPICE, Scrum or safety in trainings, lectures and his daily work. Thereby compliances and contradictions are his main focus. Timo is a member of the intacs® Advisory Board. Timo.Karasch@methodpark.de

8 Appendix A – Understanding Process Capability Levels (CL)

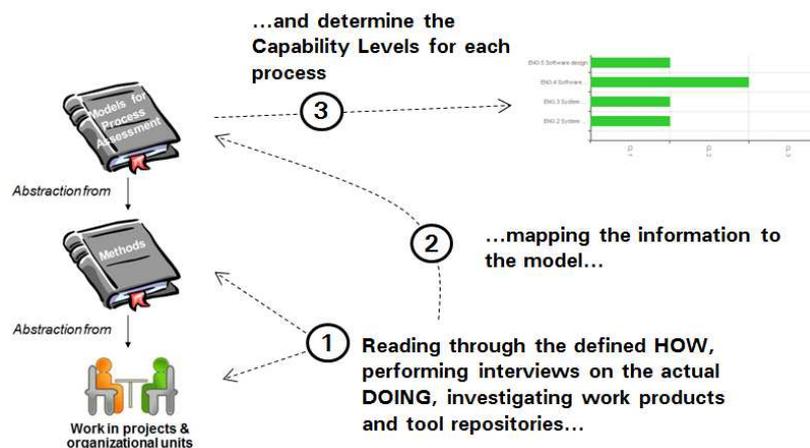


Figure 3: General idea of performing process assessments

Each individual process receives its own individual Capability Level as described below (for process-specific interpretations of those Capability Level see [6]):

Capability Level 0 vs. 1 (Performed)

CL 1 is granted if a process does fulfill its purpose, i.e. its expected outcomes are established. The outcomes are evidenced by complete work products with adequate content no matter “how this is achieved”. However, at CL1 there may still be problems such as frequent late deliveries of content, lack of qualification and training for individuals, the project being in a state of permanent reaction instead of acting proactively, and there is not always enough time for reflecting on how to do better. The notion of “no matter how the outcomes are achieved” may in fact refer to individuals working at weekends, or even “heroes” and “firefighters” who are in potential danger of burning out, or whose motivation and loyalty will suffer in the long run. Therefore, the success of CL 1 is not necessarily repeatable.

Capability Level 2 (Managed)

The organizational culture shifts from rewarding “heroes” to rewarding people who are qualified and manage to maintain discipline even in presence of operational pressure (time/budget).

Based on existing qualified/trained resources (personnel, logistics, budget, schedule, infrastructure, technology, etc.) the creation of the outcomes is now planned and effort is recorded. Quality assurance on the work done is carried out related to content and structure, work products are versioned for traceability and under configuration (what product content has been delivered to which customer, and to which product part does a customer-reported defect apply?). Responsibilities of team members for the creation of work products as well as for supporting activities are known. As a result, there is no repeated work or forgotten tasks, less time wasted for unnecessary tasks (e.g. searching for contact persons, work products, or other information, repeatedly eliminating “the same” bugs). Personnel are granted more time to learn and reflect.

In other words: sufficient and pragmatic steering and work product quality exist, however in different methodological ways across projects.

Capability Level 3 (Established)

Now there is a set of mandatory standard procedures and workflows (specific to different project sizes, customers or product families etc.). These procedures and workflows are obligatory across projects, it is however expected and wanted that they are tailored to the actual needs and context of the specific project. The standards are improved in an evolutionary way e.g. by actively collecting feedback from the performers (“improvement feedback loop”). The adapted standards are again published and trained in the organization. Consequently, projects benefit from positive experiences and mistakes already made by others. Efficient setups and conduct of new projects (people already know tools, templates, information structure) and actual reuse of parts of (work) products become possible due to the standardized way of working.

CL 3 means that the success of projects and the company does no longer solely depend on individuals but that there is a “corporate knowledge”, a “corporate memory”. A culture has evolved to follow a value-driven way of working instead of individual opinions only.

Capability Level 4 (Predictable)

Because of the standardized way of working it is now possible to actually compare process performance data delivered from the various departments and projects (you cannot compare what is not based on the same principles). Therefore the management derives information needs from the company's business goals (e.g. reducing waste, identifying efficiency potentials). Metrics (also often called KPI) are set up for feeding these information needs, data for these metrics is collected from the organizational units and projects. This data is not only used for reporting purposes; it is archived together with experienced associated effects and phenomena. Out of this data, history thresholds for “normal” and “inacceptable” can be derived. Statistical analysis can then provide “predictability”, i.e. if for a present piece of data the

data history statistics say that for this data mostly effect x has occurred then it is likely that effect x will reoccur. This means it is possible to take proactive action instead of just having to react. This replaces the “gut feeling” and enables more objective insights into observed phenomena within the organization. In one sentence: the goal of CL4 is to contribute to a proficient multi-project management and a proactive objective economic controllability of the organization.

Capability Level 5 (Optimizing)

Now data is no longer used for just “predicting potential problems” and “proactive reaction” but future occurrence of bad data is prevented through causal resolution analysis. Reasons for deviations are identified and eliminated by means of targeted process improvement (e.g. more training, new tools, more resources, organizational restructuring, technology change etc.). This includes the capability of detecting whether improvement actions were advantageous or disadvantageous.

9 References

- [1] intacs standard training course materials used worldwide for qualification of certified ISO/IEC 15504 SPICE Provisional and Competent Assessors, jointly created by Brose, BusinessCube, ISCN, Kuglermaag, Neusoft, MBTech, MethodPark, SQS, Synspace (alphabetical order), www.intacs.info
- [2] Manifesto for Agile Software Development
<http://agilemanifesto.org>
- [3] U.S. English Scrum Guide™ (July 2013)
<https://www.scrum.org/Portals/0/Documents/Scrum%20Guides/2013/Scrum-Guide.pdf>
- [4] Kanban Tutorial by Henrik Kniberg
https://www.jfokus.se/jfokus10/preso/jf-10_KanbanALeanApproachToAgileSoftwareDevelopment.pdf
- [5] DIN EN ISO 9001:2008
http://www.iso.org/iso/home/standards/management-standards/iso_9000.htm
- [6] Pierre Metz, Synspace, “Tutorial: Process-specific Interpretations of Capability Levels 2 and 3”, International SPICE Days 2009, Stuttgart, Germany