

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/221219267>

The Application of ISO 9001 to Agile Software Development

Conference Paper · June 2008

DOI: 10.1007/978-3-540-69566-0_30 · Source: DBLP

CITATIONS

21

READS

22,341

2 authors:



Tor Stålhane

Norwegian University of Science and Technology

152 PUBLICATIONS 1,427 CITATIONS

[SEE PROFILE](#)



Geir Kjetil Hanssen

SINTEF

92 PUBLICATIONS 1,636 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



TrustMe [View project](#)



Cost-efficient methods and processes for safety relevant embedded systems - CESAR [View project](#)

The application of ISO 9001 to agile software development

Tor Stålhane¹, Geir Kjetil Hanssen^{1,2}

¹The Norwegian University of Science and Technology

²SINTEF ICT

Abstract. In this paper we discuss how to reconcile agile development's focus on speed and lean development with ISO 9001's need for documentation, traceability and control. We see no need to change neither ISO 9001 nor the agile concept. Instead, we see a need to be flexible when using terms such as planning and evidence of conformance. It is true that we can include everything in agile development by making it a requirement but there is a limit to how many documents we can require from an agile process without destroying the very concept of agility.

Keywords: quality assurance, agile software development, ISO 9001

1. Introduction

With the quick advance of agile methods, some developers feel that ISO 9001 and other quality assurance standards have become irrelevant or not needed any more. The idea seems to be that an ISO 9001 conformant process is incompatible with an agile development process. Our goal is to show that there in reality is more that unite than that separate the two strategies and that both will bring benefits to a project if they are combined.

Many potential customers require that the development company has an ISO certificate before they will award it a contract. This holds both for government agencies and for private companies. The main reason for this is the level of trust created by an ISO 9001 certificate. It is much easier to check that the company has an ISO 9001 certificate than it is to check that they have a good development process and, if they have one, that they really follow it. In addition, there are many companies that are already ISO 9001 certified and want to keep their certificate while at the same time be able to introduce agile development.

2. Related work

The idea of reconciling agile development and ISO 9001 is probably almost as old as agile development itself. Even though the inventors of agile development did not consider this a problem, quite a lot of managers and quality assurance persons did.

The papers published in this area are many and varied. A problem is that some of the authors do not understand the ISO 9001 or are not aware of the fact that the standard has changed to a process oriented view with the new ISO 9001:2000. A case in point is a paper by Mnkandla and Dwolatzky [1]. Their main argument all too often boils down to statements like “the application of Object-Oriented design principles lead to maintainable systems”.

Another simple but in this case workable solution is suggested by Namioka and Bran [15]. By looking at each time box or increment as a separate project, the problem of making the process ISO 9001 conformant disappears. This solution will, however, create some extra time boxes that are only concerned with developing documentation.

McMichael and Lombardi discuss problems pertaining to aligning ISO 9001 and agile development [2] in a paper from 2007. Their main claim is that XP and Scrum together will fulfill all of ISO 9001’s requirements. Their discussion is a bit sketchy, but they are on the right tack when they state that “ISO 9001 does not equate quality. It simply helps ensure that your agile practices are being followed”. Boehm and Turner point their fingers at the same problem in [3] when they discuss the need to balance agility and discipline and observe that “Every successful venture in a changing world requires both agility and discipline”.

Vriens [4] has published a paper where he discusses the full range of CMM, ISO 9001 and their relationships to XP and Scrum. He observes that most of the ISO 9001 requirements are independent of development methods used and are covered by the existing processes.

One author who has done a really thorough job on agile development and ISO 9001 is Wright [5]. He has used an approach that has many ideas in common with the approach that we will use later in this paper – go through the ISO 9001 requirements item by item and see what XP and Scrum have to offer in line of conformance. We do, however, disagree with some of his statements and the overall conclusion that none of XP’s practices needed to be changed. We will look at two points that underline some of the problems with agile development when it comes to ISO 9001 – one is taken from the table on XP versus ISO 9001 and one is taken from the table on XP versus TickIT [6].

In his ISO 9001 versus XP table Wright has looked at the ISO 9001 item 7.3.4 “At suitable stages, systematic reviews of design and development shall be performed in accordance with planned arrangements”. The author claims that pair programming is a continuous code review. This claim does, however, not hold up against most of the available definitions of a code review – see for instance [7]. The design is not *systematically* reviewed in pair programming since the focus is on the other person’s coding. In addition, pair programming does not include documentation, which makes a later review difficult.

In Wright’s table of TickIT versus XP, he claims that “customer stories and acceptance tests fully define the software requirements”. There are two problems with this statement. Firstly, that the acceptance test defines the requirements is manifestly wrong. The acceptance test is written based on the requirements, not the other way around, although the new trend of automated acceptance testing may change this [27]. Secondly, the customer stories are way too imprecise to serve as requirements. It is

the stories *plus* the customer's acceptance – often not in writing – that define the requirements.

An approach similar to the one used by Wright is used by Ilieva et al [16]. They had a process that was already ISO 9001 conformant. Their problem was to identify how they could change the process in an agile direction and still stay ISO 9001 conformant. They called this a gap analysis and the approach seemed successful – they introduced agile development in e-business development and management to stay ISO 9001 conformant.

Melis et al. [9] focus on part seven of the ISO 9001 – product realization, since this is the part of the ISO 9001 that it most heavily touched by agile development. The paper gives a good overview of the relation between agile development and ISO 9001 for this part of the standard but leave the rest untouched. The authors identify ISO 9001 items 7.3.2 – 7.3.7 as the most important challenges for making an ISO conformant agile process.

Keenan [10] has studied ISO 9001 and XP in order to use ideas from both in a process tailoring project. He states that “the desire to support an agile development philosophy is one of the main motivators” for looking at process tailoring.

A paper by Nawrocki et al. [11] is important because the authors have performed an experiment with XP and parts of ISO 9001. The main results, as reported in the paper, are that the XP projects in the experiment suffered from such problems as low maintainability and late delivery. In addition to part seven of the ISO 9001 standard, Nawrocki also studied the effect of agile methods on part eight - measurement, analysis and improvement.

The TickIT International has also looked into ISO 9001 and agile development. Southwell sums up his observations as follows [12]: (1) many of these principles address issues which are not really covered by ISO 9001 and TickIT and are therefore not in conflict with them, (2) several principles addresses similar concerns to those of ISO 9001 but goes further and (3) some of the principles are in complete agreement even if the approaches are rather different.

We have also found two master theses [13, 14] that treat the problem of agile vs. ISO 9001. Both contain reports from case studies, which make their works important. In addition, they have done a complete review of the ISO 9001 requirements. Their goal, however, was not to check the additions needed in agile development to stay conformant to ISO 9001 but to see how well the agile development projects in the case studies adhered to ISO 9001.

Vitoria, in [13] looks at the whole TickIT standard and analyzes how it has been used in two case study projects. For the two projects in question he found that 33% of TickIT could not be applied in an XP project, 24% could be partly applied, 20% could be applied in full, while 23% were not relevant since the two projects were student project.

Erharuyi [14] looks at part seven and eight of ISO 9001, just as Nawrocki [11]. As should be expected, his conclusions are pretty much the same as those of Nawrocki. However, his paper contains some blatant misunderstandings, such as the claim that test plan updates is part of corrective actions – ISO 9001, item 8.5.2.

Another interesting case study is presented by Stephen Sykes in [17]. This case study includes an auditor's report of all findings when auditing a company using the agile method Crystal. The main conclusion is that all nonconformities identified

during the ISO 9001 audit can be solved with a little flexibility from all parties involved.

3. Agile development

Agile software development is a way of organizing the development process, emphasizing direct and frequent communication – preferably face-to-face, frequent deliveries of working software increments, short iterations, active customer engagement throughout the whole development life-cycle and change responsiveness rather than change avoidance. This can be seen as a contrast to waterfall-like processes which emphasize thorough and detailed planning and design upfront and consecutive plan conformance. Over the past ten years or so agile methods have gained great interest and popularity as they seem to address recurring problems such as budget overruns, delivering the wrong features and generating a lot of overhead in the form of reporting, formalism, re-planning and extensive management. The basic concepts of agile software development are concisely described in the agile manifesto¹. Agile software development can be seen as a philosophy and several defined methods based on these ideas are in use, all sharing a common set of values and principles. The best known and most used agile methods are Extreme Programming (XP) [22] and Scrum [23].

The main constructs used in agile development are:

- Iteration: a short (2-4 weeks) period of analysis, design, development and testing. In Scrum, iterations are called sprints.
- Product backlog: a list of prioritized requirements for the product
- Sprint or iteration backlog: a selection of items from the product backlog being developed in an iteration
- Sprint review: an evaluation of the outcome of a sprint, done in cooperation with the customer to identify fulfilled requirements and requirements needing further improvement. This can also be viewed as a retrospective. Thus, the term review here refers to review of the software being developed.
- Sprint planning: is done in the start of an iteration or a sprint and results in a sprint backlog with items that in total can be developed within the timeframe of an iteration by the current team of developers.
- Standup-meeting: a daily short meeting where each team member reports on progress, plans and problems. This can include both product and process-related problems.

¹ www.agilemanifesto.org

Compared to a strict water-fall model, an agile process involves and engages the customer both initially, in each iteration and in the finalization of the product. In each iteration the customer collaborates with the development team for requirement specification, knowledge transfer and acceptance testing – see Fig. 1.

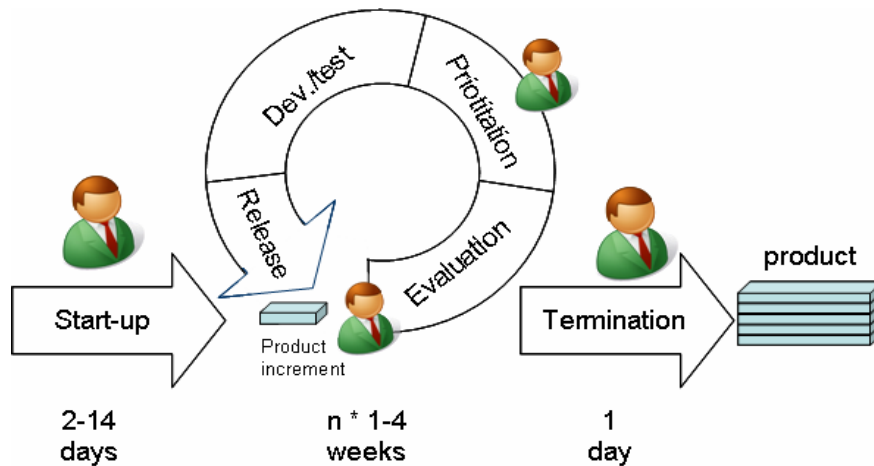


Fig. 1 Customer involvement in agile development

The ideas behind agile software development are not new [19] as they clearly are inspired by agile and lean manufacturing which have been in use in many types of industries for decades, the radical innovations in the Japanese post-war industry is probably the best known example [20]. Yet, some important changes need to be made to make this fit software development [21]. The most fundamental principle from lean development being applied is the principle of waste reduction: all work and work products not directly contributing to the development of software should be considered as waste and thus avoided or minimized.

Since the first book on Extreme Programming by Kent Beck was published in 1999 the interest and industrial use has grown surprisingly fast. The huge interest seen in industry does in most cases stem from the developers and can be explained by the simple and human-centric values carried out by agile methods which may be appealing to practitioners but threatening to management. The basic principles are easy to grasp and seems to address the most fundamental problems bugging developers. However, among this interest and willingness to radically change the development process, several critical voices have emerged and many experience reports indicate that it is not straight forward - in most cases it is an act of balancing agility and discipline [24].

One type of critique against agile methods is the deliberate avoidance of documentation as this may be considered as waste; documents are not software and software development should develop software, not documents. This is a strict concept and probably explains the common perception that agile software

development is incompatible with well known quality assurance standards such as ISO 9001. As we will show later, this does, however, not necessarily have to be true.

4. The ISO 9001 requirements

The requirements of ISO 9001 are, at the top level, summarized in a few points:

- The company must have a quality assurance management process – part 4.
- The product’s quality is the responsibility of the management – part 5. As a result of this, the company’s management must make the necessary resources available for quality assurance work and training – part 6.
- The company shall have one or more documented processes for product realization – part 7. The process must produce documents that can be (1) reviewed for acceptance and (2) used as proof of conformance.
- All reports of non-conformances, both of the product or the process, shall be reported and analyzed and should lead to a corrective action – part 8.

In addition to ISO 9001, the document ISO 90003 is also important. This is not a standard but a guideline for applying ISO 9001 to software development and maintenance.

ISO 9001 focuses on situations where we have or are about to sign a contract. The contract is signed on the basis of a defined process. One of the roles of this process is to give the customer confidence that the quality will be as specified and it is thus important that the process is followed.

ISO 9001 uses document review as its main control and verification mechanism. Many companies that use agile development claim that pair programming makes reviews unnecessary. In addition, there exist both experiments and case studies that show that pair programming, given the right conditions, give a lower failure rate than traditional software development – see for instance [18, 28]. We can, however, not dispose of document reviews and still claim ISO 9001 conformance. A statement from “Det norske Veritas” is enlightening here: “The main point is that verification shall be performed according to plan – see ISO 9001, item 7.3.5. The amount of verification needed must be adapted to the importance of each artifact and our confidence in the process that produced it. If the company has developed a strong confidence in the results from e.g. pair programming, it will be reasonable to move the resources somewhere else where there is a larger probability that the verification process will contribute to a better product quality. Thus, pair programming will influence the verification plan but cannot be used as an argument to suspend verification and verification planning”.

5. Comparing ISO 9001 with agile development

When comparing ISO 9001 and agile development, we have defined agile development as adherence to the basic principles as formulated in [19]. In order to

compare agile development with ISO 9001, we used the same approach as we have earlier used when introducing ISO 90001 to a Norwegian company [8]. The approach is simple – use a table containing all the ISO 9001 items aligned with the corresponding item in the ISO 90003 guidelines. Have a free column where we can indicate whether agile development meets the requirements of ISO 9001 as explained in ISO 90003. All items not ticked off are given a closer scrutiny to see whether we need to add or change something in order to achieve conformance.

In addition to this, we compared our assessment to the assessments in [5], [13], [14] and [17] and did a closer investigation where there was a disagreement. This process left us with 15 items where agile methods only partly were able to fulfill the ISO 9001 requirements and four items where agile methods could not meet the requirements at all. As should be expected, part seven of the ISO 9001 standard dominates in both cases – nine out of 15 of the partly fulfilled items and two of the four items that were not fulfilled at all. On the positive side – of the 50 items in ISO 9001, 31 items will not need any changes or enhancements whatsoever by either side.

The results of the comparison process were used as our starting point for a three step process: (1) identify the reasons for the lack of full conformance, (2) see how these lacks can be amended, either by extending the development process or by augmenting the ISO 90003 guidelines.

6. What can be done to achieve conformance

4.2.1.d: The quality management system documentation shall include documents needed by the organization to ensure the effective planning, operation and control of its processes.

As for any kind of development methodology, an agile development project always starts by defining how the methodology shall be used in the given project. The planning of an agile project can easily be documented in a simple form specifying for instance iteration length, how to record and track requirements etc.

4.2.4: Records shall be established and maintained to provide evidence of conformity to requirements and of the effective operation of the quality management system. Records shall remain legible, readily identifiable and retrievable. A documented procedure shall be established to define the controls needed for the identification, storage, protection, retrieval, retention time and disposition of records.

In between iterations, conformity to requirements are evaluated by the product owner and records of the results are kept as evidence of conformity.

5.3: Top management shall ensure that the quality policy

- *5.3 a: is appropriate to the purpose of the organization*
- *5.3 b: includes a commitment to comply with requirements and continually improve the effectiveness of the quality management system*

A well implemented agile process, with the necessary conditions in place will support the purpose of the organization. That is, delivering well functioning software within the compromised time- and cost frame. A well working agile process will

ensure a dedicated commitment to requirements as these are continuously evaluated based on experience from development and testing.

5.4.1: Top management shall ensure that quality objectives, including those needed to meet the requirements for the product, are established at relevant functions and levels within the organization. The quality objectives shall be measurable and consistent with the quality policy.

The agile method Evo [29] emphasizes measurable quality objectives and this practice can easily be applied in other agile methods and meets the ISO 9001 requirement.

5.6.2: The input to management review shall include information on

- *5.6.2 a: results of audits*
- *5.6.2 b: customer feedback*
- *5.6.2 c: process performance and product conformity*

After each iteration, the process performance is reviewed and potential improvements are implemented in the following iteration. Reviews are based on input from developers and customer feedback. Output from such retrospectives [25] can in the context of ISO 9001 be used as input to management review.

7.1: The organization shall plan and develop the processes needed for product realization. Planning product realization shall be consistent with the requirements of the other processes of the quality management system (see 4.1). In planning product realization, the organization shall determine the following as appropriate

7.1a: quality objectives and requirements for the product

7.1 b: the need to establish processes, documents and provide resources specific to the product

The adaptation of an agile process at the start of the development project covers this requirement. Adaptation, or process planning, may include deciding iteration length, strategies for requirements documentation, staffing etc.

7.2.1a: The organization shall determine requirements specified by the customer, including the requirements for delivery and post-delivery activities.

Agile processes include practices for determining requirements but this is usually focused on features and qualities of the product itself. To cover the requirement of determining requirements for delivery and post-delivery activities this needs to be included. One way of dealing with this is to add additional sprints after delivery to deal with any post-delivery activities.

7.2.2: The organization shall review the requirements related to the product. The review shall be conducted prior to the organization's commitment to supply a product to the customer (e.g. submission to tenders, acceptance of contracts or orders, acceptance of changes to contracts or orders) and shall ensure that

- *7.2.2a: product requirements are defined.*
- *7.2.2 c: the organization has the ability to meet the defined requirements*

- 7.2.2 x2: *where the customer provides no documented statement of requirements, the customer requirements shall be confirmed by the organization before acceptance*

This ISO 9001 requirement is the most problematic if it is to be interpreted strictly. It requires the development organization to have the complete requirements defined upfront. However, agile methods actually do specify that requirements should be gathered upfront. They will, however, not be complete and will not contain all the details. This is based on the assumption that it is impossible to get a complete overview of all details up front; instead the most important aspects should be documented. In Scrum this is documented in the product backlog which is set up prior to the first iteration - at this time it constitutes the best possible understanding of the requirements. Compared to traditional requirements specifications it differs in the way that it is anticipated to change, based on experience from development. The conclusion of this issue is that if an auditor accepts this initial overview of requirements, agile methods fulfil this requirement. If not, we find that the fundamental principle of requirements evolution in agile methods is in conflict with ISO 9001.

7.3.1a: The organization shall plan and control the design and development of the product. During the design and development planning, the organization shall determine the design and development stages.

Agile methods cover planning and control of the product design. It differs from the traditional approach in that this is done iteratively and incrementally, yet it is handled. It will still produce documents that can be used as proof of conformance for the activities mentioned in 7.3.1a. Examples of documents that have been accepted as proof of conformance are e.g. pictures of the whiteboard showing requirements planned, in work or completed.

7.3.2: Inputs related to product requirements shall be determined and records maintained (see 4.2.4). These inputs shall include

- 7.3.2 a: *functional and performance requirements.*
- 7.3.2 x1: *these inputs shall be reviewed for adequacy. Requirements shall be complete, unambiguous and not in conflict with each other*

This is handled through the requirements process in agile methods. In front of each iteration new requirements are gathered or existing requirements altered due to customer feedback. These requirements are then reviewed and recorded through cooperation between the customer and the development team. This includes both functional and performance requirements.

7.3.3: The outputs from design and development shall be provided in a form that enables verification against the design and development input and shall be approved prior to release. Design and development output shall

- 7.3.3 a: *meet the input requirements for design and development*

There seems to be a certain amount of disagreement in the agile camp as to whether e.g. XP requires design but all this aside, it is no problem to include a high level design activity in the first planning game and a low-level design at the start of each iteration.

7.3.4: At suitable stages, systematic reviews of design and development shall be performed in accordance with planned arrangements (see 7.3.1)

7.3.4 a: to evaluate the usability of the results of design and development to meet requirements

Design and development is reviewed in every transition between iterations as a joint effort between the development team and the customer. The customer is given a particular responsibility of evaluating usability and requirements conformance.

7.3.5: Verification shall be performed in accordance with planned arrangements (see 7.3.1) to ensure that the design and development outputs have meet the design and development requirements. Records of the results of the verification and any necessary actions shall be maintained

It is not a problem to include a high level design activity in the first planning game and a low-level design at the start of each iteration.

7.3.7: Design and development changes shall be identified and records maintained. The changes shall be reviewed, verified and validated, as appropriate, and approved before implementation. The review of design and development changes shall include evaluation of the effect of the changes on constituent parts and products already delivered. Records of the results of the review of changes and any necessary actions shall be maintained (see 4.2.4)

The review done after each iteration takes care of this. This is done as a joint effort between the development team, the customer or product owner and other possible stakeholders which amongst other issues consider changes to design and development. Decisions are documented in the form of an updated product backlog. If necessary, formal and signed minutes of meeting can be made to keep track of the design and development history.

8.1: The organization shall plan and implement the monitoring, measurement, analysis and improvement processes needed

- *8.1 a: to demonstrate conformity of the product*
- *8.1 b: to ensure conformity of the quality management system*

This is covered by the planning and adoption of the agile method being used. A central part of all agile methods is close monitoring of progress to early discover potential problems. The reviews done between iterations also include an evaluation of the development process itself to potentially identify software process improvement initiatives.

8.2.3: The organization shall apply suitable methods for monitoring and, where applicable, measurement of the quality management system process. These methods shall demonstrate the ability of the process to achieve planned results. When planned results are not achieved, correction and corrective action shall be taken, as appropriate, to ensure conformity of the product.

Besides evaluation of product increments, the review in between iterations also may include a retrospective [25]. This has the same function as a traditional

assessment of the process performance, potentially leading to process improvement actions to be implemented in the following iterations.

8.2.4: The organization shall monitor and measure the characteristics of the product to verify that product requirements have been met. This shall be carried out at appropriate stages of the product realization process in accordance with the planned arrangements (see 7.1)

Evidence of conformity with the acceptance criteria shall be maintained. Records shall indicate the person(s) authorizing release of product (see 4.2.4)

This is handled by the iteration reviews. Acceptance of requirements are documented e.g. in the product backlog or similar.

8.5.2: The organization shall take action to eliminate the cause of nonconformity in order to prevent recurrence. Corrective actions shall be appropriate to the effects of the nonconformity encountered.

A documented procedure shall be established to define requirements for

- *8.5.2 c: evaluating the need for actions to ensure that nonconformities do not recur*
- *8.5.2 d: determining and implementing action needed*
- *8.5.2 e: records of the results of action taken (see 4.2.4)*
- *8.5.2 f: reviewing corrective action taken*

The iteration reviews intend to discover nonconformity with requirements. This is either caused by too little resources in the previous iteration due to unforeseen difficulties, insufficient understanding of the requirements or a bad process. Only the latter case is of interest here. The process causes are registered and will later be used as input to a process improvement activity.

8.5.3: The organization shall take action to eliminate the cause of potential nonconformities in order to prevent their occurrence. Preventive actions shall be appropriate to the effects of the potential problems.

A documented procedure shall be established to define requirements for

- *8.5.3 a: determining potential nonconformities and their causes*
- *8.5.3 b: evaluating the need for action to prevent occurrence of nonconformities*
- *8.5.3 c: determining and implementing action needed*
- *8.5.3 d: records of results of action taken (see 4.2.4)*
- *8.5.3 e: reviewing preventive action taken*

The intention of having frequent reviews of development progress, requirements and process performance in cooperation with the customer is, among other things, to eliminate causes of potential nonconformities. As both the software product and related knowledge grows, the development team and the customer continuously improve their ability to discover potential sources of nonconformity. Such reviews may, if relevant, produce software process improvement actions – both to reactively take immediate action and as a mean to proactively improve the development process for later development projects.

7. Threats to validity

There are three threats to validity for our conclusion – have we (1) understood ISO 9001, (2) have we touched all relevant ISO 9001 items and (3) have we understood agile development in general and Scrum and XP in special?

7.1 Have we understood ISO 9001

One of the authors has experience with helping a company becoming ISO 9001 certified and has a through knowledge of ISO 9001. Whenever we have been in doubt, we have consulted personnel at DnV who certify Norwegian companies. They have a large amount of ISO 9001 experience and have been able to clear up any misunderstandings that we might have had.

An ISO 9001 certification audit is, however, not an exact science. Different auditors may have different standards for what they find acceptable. Thus, there is always a possibility that what we have found acceptable – e.g. the Scrum planning process – may not be accepted as a planning process by some auditors.

7.2 Have we touched all relevant ISO 9001 items

By using the standard itself plus its guideline, we went through the whole standard, item by item. All items that concerned documents, documentation or acceptance of documents, together with all issues pertaining to the implementation, validation and verification of a software system were assessed – see chapter 5.

In addition, we have coordinated our findings with four independent sources and all ISO 9001 items we identified were also identified by at least one of these sources. Thus, we are confident that all relevant ISO 9001 items are identified and assessed.

7.3 Have we understood agile development

Agile development is not an exact defined methodology and there exist a handful of agile methods that varies [26]. Yet they are all based on the few common principles described in chapter 3. In our assessment we have tried to apply these common and fundamental principles to reduce a potential bias from our own interpretations of what agile development is, yet a certain level of subjective interpretation is inevitable.

7.4 Our claims to validity

Based on the discussion in the sections 7.1 to 7.3 we claim that our conclusions regarding ISO 9001 and agile development will be valid for a wide range of companies and auditing authorities.

8. Conclusion and future work

Based on the discussions in the chapter on threats to validity, we feel that our observations and conclusions are relevant for the topic.

The main difference between ISO 9001 and agile methods is that ISO 9001 insists on documentation for reviews and to demonstrate process conformity. Agile methods try to avoid writing documents that does not contribute to the finished system. On the other hand – if the customer requires a certain document, the use of agile methods are no hindrance for developing them.

There are ways to deal with many of the documents that ISO 9001 requires. We can add such activities as review meetings, writing design documents and so on. The process will still keep the most important agile ideas, such as short iterations, building in increments, including the customer, reprioritizing requirements whenever need, and constantly adjusting scope, time and cost within the bounds of the project contract. One often used slogan in the agile community is “Do the Simplest Thing that Could Possibly Work”. The term “simplest” does not mean it is forbidden to add extra process artifacts or activities. There are, however, limits to how many new artifacts that can be added to an agile method and still insist on labeling it agile. The changes necessary to be conformant to ISO 9001 are, however, well inside those limits.

We see from the discussions above that the differences between agile development and an ISO 9001 conformant development process are not insurmountable. Some changes are, however, needed. We suggest the following actions:

ISO: the ISO 90003 guidelines should include some guidelines concerning (1) what is accepted as a review (2) several types of reviews and (3) when each type of reviews is considered necessary.

Agile development: given the suggestions to the ISO 90003 guidelines, there remains two problems – that an agile process produce documents that can be used (1) as proof of conformance and that (2) can be reviewed as part of ISO 9001’s verification and validation.

When the abovementioned problems are solved, there will be no problems whatsoever when a company wants to use agile development and still keep its ISO 9001 certificate.

Acknowledgements

We gratefully acknowledge the help from T. Skramstad and F. Prytz, DnV for important input and fruitful discussions on the interpretation of ISO 9001.

References

- [1] Mnkanla, E. and Dwolatzky, B. “Defining Agile Quality Assurance” in the proceedings of the International Conference on Software Engineering Advances – ICSEA’06, 2006
- [2] McMichael, M. and Lombardi, M. “ISO 9001 and Agile development”, AGILE 2007

- [3] Boehm, B. and Turner, R. "Balancing Agility and Discipline: Evaluating and Integrating Agile and Plan-Driven Methods" in the proceedings of 26th International Conference on Software Engineering – ICSE'04 – 2004.
- [4] Vriens, C. "Certifying for CMM Level and ISO 9001 with XP@SCRUM", in the proceedings of the Agile development Conference – ADC'03, 2003.
- [5] Wright, G. "Achieving ISO 9001 Certification for an XP Company" in the book XP/Agile Universe 2003, LNCS 2753, Springer-Verlag Berlin Heidelberg 2003.
- [6] The TickIT Guide, British Standards Institute, London, UK, 2001.
- [7] IEEE "Standard Glossary of Software Engineering Terminology", IEEE Std 610.12-1990, Standards Coordinating Committee of the Computer Society of the IEEE
- [8] Stålhane, T. "Implementing an ISO 9001 certified process", proceedings of the EuroSPI Conference, Joensuu, Finland, 2006
- [9] Melis, M. et al. "Requirements for an ISO Compliant XP Tool", XP 2004 LNCS 3092, Springer-Verlag Berlin Heidelberg 2004.
- [10] Keenan, F. "Agile Process Tailoring and Problem Analysis (APTLY)", in the proceedings of 26th International Conference on Software Engineering – ICSE'04 – 2004.
- [11] Nawrocki, J.R, et al.: "Combining Extreme Programming with ISO 9000"
- [12] Southwell, K. "Agile Process Improvement", in TickIT International, ISSN 1354-588, published by Firm Focus on behalf of BSI-DISC
- [13] Vitoria, D.: "Aligning XP with ISO 9001:2000 – TickIT Guide 5.0 – A case study in two academic software projects", Master Thesis, School of Engineering, Blekinge Institute of Technology, Ronneby, Sweden, 2004
- [14] Erharuyi, E.: "Combining eXtreme Programming with ISO 9000:2000 to Improve Nigerian Software Development Processes", Master Thesis, School of Engineering, Blekinge Institute of Technology, Ronneby, Sweden, 2007
- [15] Namioka, A. and Bran, C.: "eXtreme ISO", proceedings of the OOPL'04, October 24-28, 2004, Vancouver, Canada.
- [16] Ilieva, S. et al.: "Analysis of an agile methodology implementation", proceedings of the 30th EUROMICRO conference, 2004.
- [17] Cockburn, A.: "Crystal Clear – A Human-Powered methodology for Small Teams", Addison-Wesley Longman, Amsterdam, ISBN 0201 699478.
- [18] Aiken, J.: "Technical and Human Perspective on Pair Programming", ACM SIGSOFT Software Engineering Notes, vol. 29, no. 5, September 2004
- [19] Merisalo-Rantanen, H. and Rossi, M.; "Is Extreme Programming Just Old Wine in New Bottles: A Comparison of Two Cases", Journal of Database Management, 2005.
- [20] Takeuchi, H and Nonaka, I.: "The New Product Development Game". Harvard Business Review, 1986
- [21] Poppendieck M. and Poppendieck T.: "Lean Software Development: An Agile Toolkit for Software development Mangers", The Software Development Series, ed. A. Cockburn and J. Highsmith, 2003, Addison Wesley.
- [22] Beck, K.: "Extreme programming explained: embrace change", 2000, Addison-Wesley
- [23] Schwaber, K. and Beedle, M. "Agile Software Development with Scrum", 2001, Prentice Hall
- [24] Boehm, B. and Turner, R. "Balancing Agility and Discipline – A Guide for the Perplexed", 2004, Addison-Wesley.
- [25] Derby, E. and Larsen, D.: "Agile Retrospectives: Making Good Teams Great", 20067, Pragmatic Bookshelf
- [26] Abrahamsson, P. et al.: "Agile software development methods – review and analysis", 2003, VTT Electronics

- [27] Mugridge, R. and W. Cunningham, "Fit for Developing Software: Framework for Integrated Tests" (Robert C. Martin). 2005: Prentice Hall PTR Upper Saddle River, NJ, USA.
- [28] Arisholm, E., et al., "Evaluating Pair Programming with Respect to System Complexity and Programmer Expertise". IEEE Transactions on Software Engineering, 2007. 33(2): p. 65-86.
- [29] Gilb, T., Competitive Engineering: A handbook for systems engineering, requirements engineering, and software engineering using Planguage. 2005: Elsevier Butterworth-Heinemann