ISSN 1330-3651 (Print), ISSN 1848-6339 (Online)
DOI: 10.17559/TV-20140704180948

GETTING ISO 9001 CERTIFIED FOR SOFTWARE DEVELOPMENT USING SCRUM AND OPEN SOURCE TOOLS: A CASE STUDY

Tomo Popović

Subject review
This article presents a case study of adopting the Scrum process for Java development using tools coming from the open source software world. The approach is focused on small to medium sized software enterprises, which can sometimes be intimidated by the introduction of formal processes and quality management systems. The paper describes agile practices, Scrum process, and tools selection aimed at software development using Java. Each selected tool is described and put into the context of the Scrum process and implementation of a quality management system. The paper introduces ISO 9001:2008 standard and discusses benefits of the implementation of a quality management system. Finally, the case study illustrates how the selected tools and Scrum artefacts can be used for the adoption of a quality management system, which leads to a successful ISO 9001:2008 certification.

Keywords: *agile development; ISO 9001; Java; open source software; software development*

Dobivanje ISO 9001 certifikacije za razvoj softvera uporabom Scrum procesa open source alata: studija slučaja

Pregledni članak
Članak predstavlja studiju praktičnog usvajanja Scrum procesa za razvoj softvera u Java programskom jeziku uz uporabu open source alata. Opisani pristup se fokusira na mala i srednja softverska preduzeća koja se u praksi najčešće opiru formalizaciji razvojnog procesa i implementaciji sustava za upravljanje kvalitetom. Ovaj članak opisuje agilni razvoj softvera, Scrum proces i izbor open source alata za razvoj na Java platformi. Izabrani alati su stavljeni u kontekst Scrum procesa i implementacije sustava za upravljanje kvalitetom. Članak opisuje zahtjeve ISO 9001:2008 standarda za upravljanje kvalitetom i diskutira prednosti njegove implementacije. Konačno, studija ilustrira kako se izabrani alati i pragmatična uporaba Scrum agilnog procesa mogu koristiti kao osnova za implementaciju sustava za upravljanje kvalitetom čime se omogućava i uspješno dobivanje ISO 9001:2008 certifikata.

Ključne riječi: *agilni razvoj softvera; ISO 9001; Java; opensource; razvoj softvera*

1 Introduction

In this paper, an implementation of a quality management system based on ISO 9001 in small to medium sized software enterprises is discussed. The paper provides a case study of ISO 9001 certification for the use of Scrum process for Java development using open source software (OSS) tools. The awareness about OSS has been around for some time and its popularity rapidly has risen over the last 25 years [1, 2]. This is due to various reasons, but mainly because of the wide availability of the Internet, increased quality of the software offered by the OSS communities, and increasing industry support. The OSS development promotes evolutionary thinking, and iterative and incremental processes, which is very much in line with agile methods for software development [3].

In the software development, we have also witnessed some dramatic changes in the approach to how to efficiently develop high quality software. More and more companies are moving away from the traditional "waterfall" approach towards iterative and incremental processes called agile development [4, 5]. Being agile means better collaboration within team members and with customers, as well as frequent software releases that allow for continuous reflection, verification, and exposure to scrutiny. The idea is to create environment for efficient navigation of the project. The focus is on the users and new value, while maintaining a sustained quality and pace of software development process [6].

The case study described in this paper illustrates implementation of the agile process called Scrum in a small software enterprise. By using only the OSS tools, this approach can successfully be implemented in a small to medium enterprises providing software development services to enable a successful ISO 9001 certification [7]. The paper is organized as follows. Related work is provided in the next section. Information on agile development, Scrum process, quality management systems per ISO 9001, and Java development is given in Section 3. The fourth section focuses on the implementation of the quality management system. This section discusses important considerations for ISO 9001 certification assuming the context of small to medium sized software teams. This discussion results in the mapping between ISO 9001 requirements and Scrum process artefacts. The discussion and conclusions are given at the end.

2 Related work

The idea of applying ISO 9001 to agile software development is most likely as old as the ideas about agile development itself [8, 9]. A good analysis of application of ISO 9001 to agile development processes is given in [10]. A really good material with detailed considerations on agile development and ISO 9001 is provided in [11], although with a main focus on eXtreme Programming (XP). Additional examples of combining XP and ISO 9001 were given in [12] and [13]. In [14], the authors propose combining Scrum and XP in order to align with ISO 9001. This is really not necessary, but the authors made a good point that implementing ISO 9001 does not equate quality, but insures that the agile practices are being followed.

There is often a confusion and misunderstanding of both agile development processes and ISO 9001 and its purpose. Sometime, the confusion comes from not knowing that the ISO 9001 changed over time and it is mainly focused on the quality management system implementation, its maintenance, and continuous improvement [15, 16]. The authors in [17 ÷ 19] explore details of applying ISO 9001:2008 to XP development process. An interesting experience is reported in a case study for a small to medium sized company in China attempting to implement CMMI after implementing ISO 9001 to their iterative process based on RUP [20]. A very useful discussion of introduction of Scrum to a CMMI Level 5 company is given in [21]. A presentation on implementing ISO 9001:2000 with Scrum provides good details and discussion on how to achieve conformance in agile fashion can be found in [22].

This paper reports on the use of Scrum in a small sized software enterprise and describes considerations and experiences with ISO 9001:2008 certification.

3 Agile, Scrum, and ISO 9001

3.1 What is Agile?

Agile software development assumes a set of lightweight development methods that enable efficient, iterative, and incremental process for development and maintenance of a software product. It promotes evolutionary thinking, adaptive process, and effective response to change. The incremental iterations are typically time-boxed and short allowing for easy process navigation, corrections, and rapid response to requirements changes. The agile processes can be recognized by understanding the values introduced in the Agile manifesto back in 2001 (Tab. 1). The items in the left column are important to the development process, but the items shown in the left column are valued more [23]. Agile processes are lean with an emphasis on collaboration and communication between the developers, users, and other stakeholders. The focus of this collaboration is bringing the new value to the user in the most efficient way.

Table 1 Values from the Agile Manifesto

More value	Important, but less value
Individuals and interactions	Process and tools
Working software	Detailed documentation
Customer collaboration	Contract negotiation
Responding to change	Following a plan

3.2 Scrum process

The Scrum process is one of the most popular agile frameworks for completing complex projects (Fig. 1). The process is defined around time-boxed iterations, called Sprints, which are typically fixed to 2 ÷ 4 weeks. Management of the process includes a continuous update of the product specifications called Product Backlog. For each iteration, the team as a whole sits together, selects a set of doable items from the product backlog, and breaks them down into tasks containing more detailed description of the work. Each day during the sprint the team gathers for short Daily Scrum meets, typically stand up meetings, where the team members inform each other of the progress and activities. The daily meetings are not meant to be reporting to superiors, but to keep everyone up to date and to expose possible obstacles that need to be resolved. At the end of the sprint, it is expected to provide a potentially shippable increment. At the Review Meeting the team demonstrates newly implemented features, fixed bugs, or other product related artefacts. The presentations, slides, and progress reports are not considered to be valuable. The emphasis is on alive demo of the new features and their value for the user. After each sprint, the team can have a Retrospective Meeting to analyse good and bad practices, and discuss possible improvements to the process itself. The process defines roles such as Product Owner, Scrum Master, and team members. The process artefacts that support collaboration and communication are Product Backlog (requirements, feature list), Sprint Backlog (tasks list, sprint to-do list), and product or sprint burn-down charts. More details on Scrum can be found in [24].

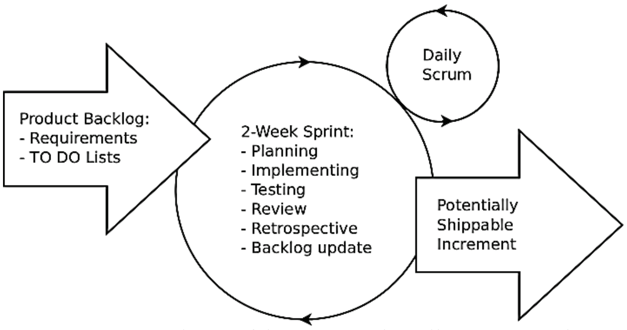

Figure 1 Scrum is one of the most popular agile processes today

User Stories have a key role in managing software product requirements [25]. The concept is to capture the description of the feature or functionality in a short one to two sentence form.Widely adopted template is: "As a <role>, I want <goal/desire> so that <benefit>". User stories capture the functionality as seen by the user and spell out the relationship between the features and benefits/values it brings to the user. The requirements are quickly defined by using user stories, and the details can be left for later conversations that take place at the time of the implementation.

3.3 What is ISO 9001?

The ISO 9001 standard is related to quality management systems (QMS) and designed to help organizations to meet the needs of their customers and other stakeholders. It is designed for organizations that build products or provide services [7, 26]. The quality management system model in ISO 9001 is depicted in Fig. 2. The involvement of the management is crucial for successful implementation of the QMS. The main goal is to achieve high customer satisfaction through consistent products. The model is set up so that it enables continual improvements of the QMS, while the products or services are being delivered. Implementing a QMS and ISO 9001 certifications are important even for small and medium software development enterprises. The benefits can be reflected in sustaining and improving the quality of the development process and software products, but also in the appearance to potential clients and for project bids that require QMS certification.

The first step towards certification is to obtain a copy of the ISO 9001:2008 standard from one of the suppliers. ISO 9001 is directly audited against by the independent, third party, certification body. Before the certificate is

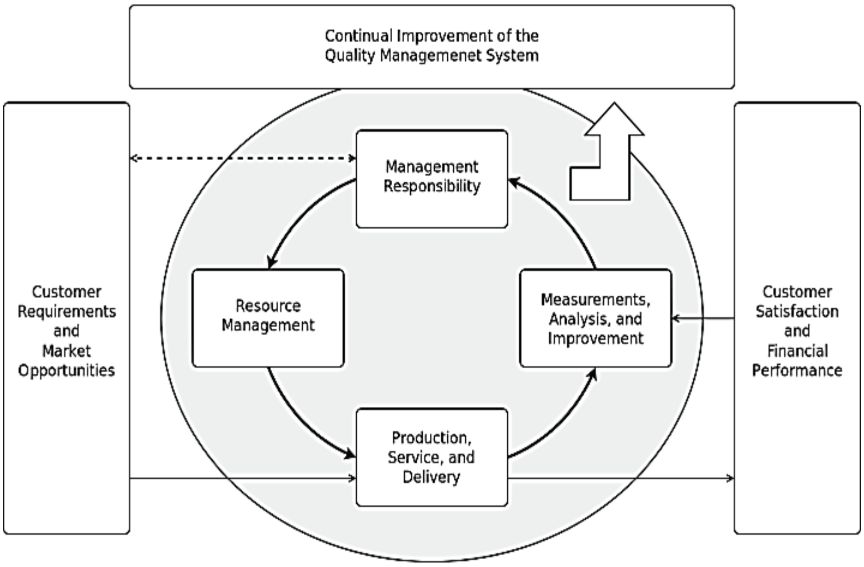
Continual Improvement of the Quality Managemenet System
Management Responsibility
Resource Management
Measurements, Analysis, and Improvement
Production, Service, and Delivery
Customer Requirements and Market Opportunities
Customer Satisfaction and Financial Performance

Figure 2 Quality management system (ISO 9001)

issued or renewed, the auditor must be satisfied that the company being assessed has implemented the requirements in sections 4 to 8 of the ISO 9001 standard. There are seven required documents that the company seeking certification needs to prepare:

- Quality Manual, definition of the quality policy and objectives (4.2.2),
- Control of Documents (4.2.3),
- Control of Records (4.2.4),
- Internal Audits (8.2.2),
- Control of Nonconforming Product / Service (8.3),
- Corrective Action (8.5.2),
- Preventive Action (8.5.3).

There is a common misunderstanding that ISO 9001 implementation of a QMS has to be document-intensive and that it cannot be achieved in a small and medium enterprise environment. The required documents can be concise and they need to be developed once for a given company. The documents are changed only when the implemented QMS evolves and these changes can be in a form of minor and incremental updates. Another misconception is that documentation is not part of agile processes, which is wrong. Although working software is the priority, the correct and necessary documentation is required. This paper will address how to map documentation artefacts of a Scrum process into the ISO 9001 requirements. Furthermore, some of the requirements can be documented using issue tracking tools, which are commonly found in software development enterprise.

3.4 Use of OSS tools for Java development

The focus in this article is on developing software products using Java programming language and related technologies [27]. There is a wide spectrum of high quality development and deployment tools for Java. The OSS tools used for the development described in this case study include the following [28]:

- Java platform (JDK, JRE)
- Integrated Development Environment (Netbeans, Eclipse)
- Version Control (Subversion)
- Configuration Management (Subversion, Maven, Archiva)
- Automated Testing (Concordion)
- Continuous Integration (Jenkins)
- Issue Tracking (Bugzilla)
- Documentation (LibreOffice, ProjectLibre).

The development setup comprised of the selected tools is depicted in Fig. 3. The backbone of the configuration is the development server, which runs Linux. The server hosts Subversion for code repository, Maven for library repository, Bugzilla for issue tracking, and dedicated space for in-house file backup. The development PC runs desktop version of Linux and development tools including OpenJDK, NetBeans, JUnit, Concordion, and others. The office PCs are used for Internet browsing and documentation editing. All of these and other programs can be installed on a single machine used by a team member. The CI server acts as another team member. It periodically checks out the latest version of the code on the development server. Jenkins performs the build process and runs the automated tests. It reports the errors to the team and posts the update on its web pages. The Linux distribution used in the given setup is Ubuntu Long-Term-Support, versions 12.04 LTS and more recent 14.04 LTS [29].

4 Implementing the QMS

The implementation of the presented approach resulted in a very interesting outcome. The pragmatic use of the selected tools and Scrum streamlined the adoption of the quality management system defined by the ISO 9001:2008 standard. This is extremely important, especially for small and medium sized software enterprises, as being ISO 9001 certified increases the

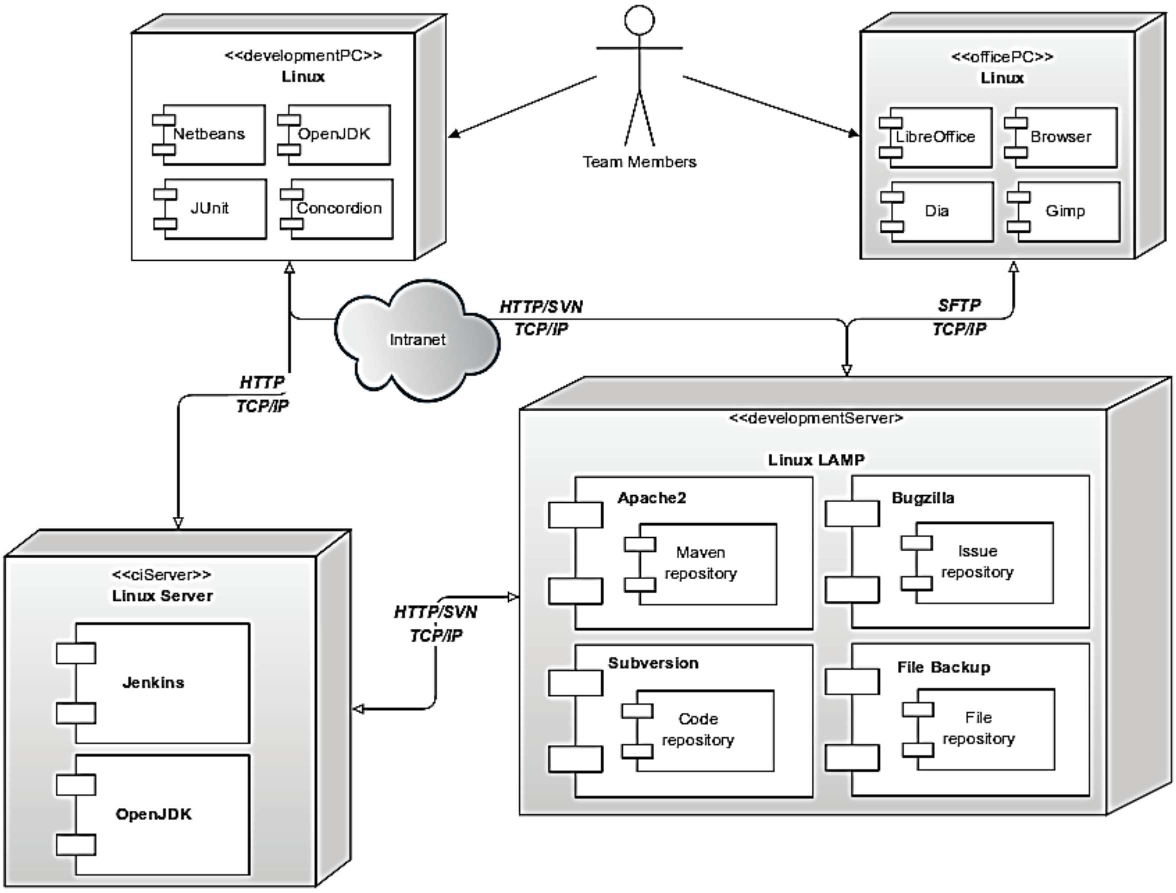

Figure 3 Deployment diagram of the development setup

visibility and acceptance by future clients. The following sections introduce the ISO 9001 and discuss mapping between Scrum process artefacts and ISO 9001 requirements.

4.1 ISO9001 requirements

For small enterprises all of this may seem intimidating, but QMS per ISO 9001:2008 is actually easier to implement than per its previous versions. It is, however, advisable to obtain the training materials, document templates, and, if needed, check with the consulting firms [30, 31]. In our case we used the templates to create polices, quality manual, and procedure documents (listed in section 3.3). Most of the information needed to be filled into the templates is fairly straightforward. The following subsections address some peculiarities that needed additional considerations and effort needed to align software development with Scrum and ISO 9001 requirements.

4.2 Organizational chart

First, it was needed to identify roles pertinent to QMS and define the organizational chart. Small to medium companies tend to be overwhelmed with this task and often opt for complicated organizational charts. It often happens that in such charts one person fills many roles, which is unnecessary complication. The key here is to keep it simple, especially for the companies that have only few employees. There is no need to complicate the organizational chart or assign multiple roles to the same individual.

4.3 Creating a process diagram

Another aspect here is to define the process diagram. For software development enterprises the company can be defined as both product and service provider. After some careful considerations, it was decided for the option to define the processes for the company as being a service provider. In this particular case, it seemed as a simpler option to treat software development as a service being provided by the company. It may not be possible for every small-medium software company out there, but it seemed more feasible reasonable in our case.

The process diagram is illustrated in Figure 4. In our case, we defined four main processes: a) Quality Management; b) Contract Management; c) Service Delivery; and d) Purchasing. It is important to note that all of the processes and their corresponding activities have to be traceable to the ISO 9001 requirements specified in sections 4-8 in the standard. The process chart in Figure 4 can be re-used by majority of small and medium software development teams.

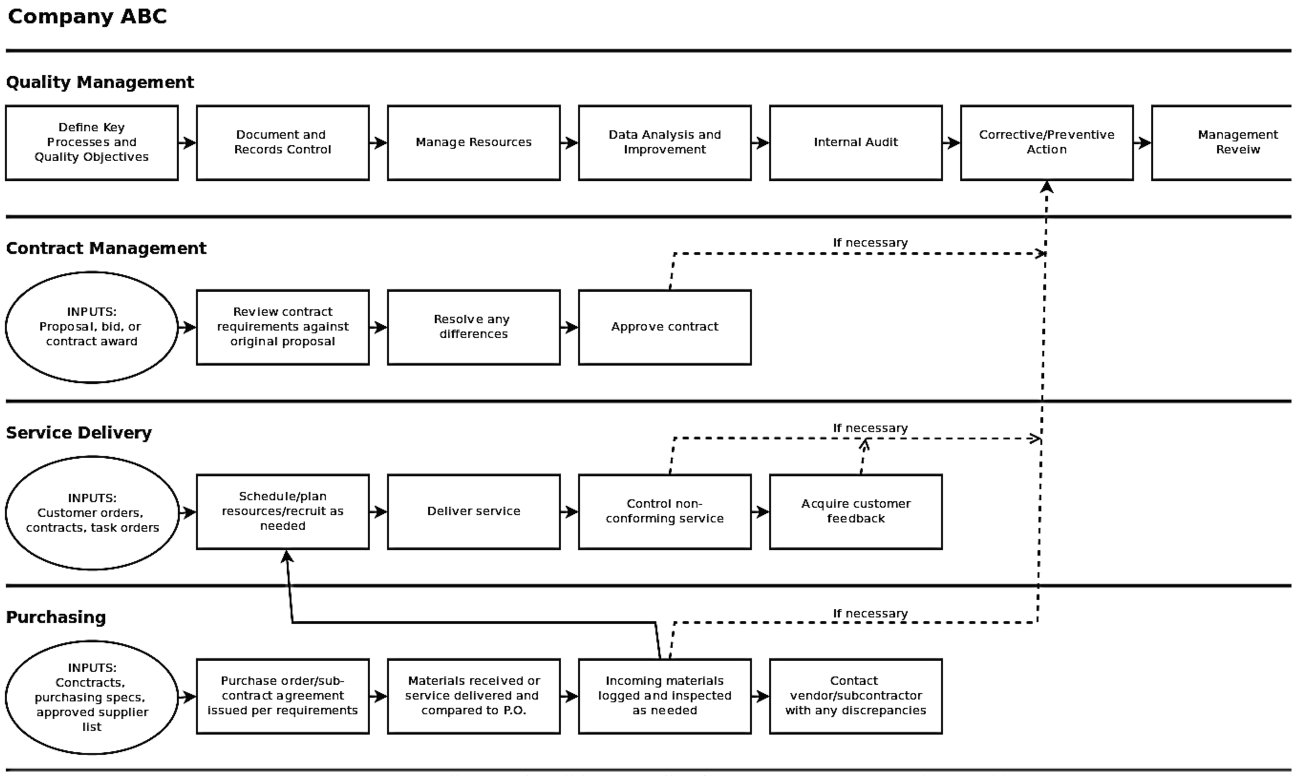
Company ABC
Quality Management
Define Key Processes and Quality Objectives
Document and Records Control
Manage Resources
Data Analysis and Improvement
Internal Audit
Corrective/Preventive Action
Management Revelw
Contract Management
If necessary
INPUTS: Proposal, bid, or contract award
Review contract requirements against original proposal
Resolve any differences
Approve contract
Service Delivery
If necessary
INPUTS: Customer orders, contracts, task orders
Schedule/plan resources/recruit as needed
Deliver service
Control non-conforming service
Acquire customer feedback
Purchasing
If necessary
INPUTS: Conctracts, purchasing specs, approved supplier list
Purchase order/sub-contract agreement issued per requirements
Materials received or service delivered and compared to P.O.
Incoming materials logged and inspected as needed
Contact vendor/subcontractor with any discrepancies

Figure 4 A process diagram describing a small software enterprise as a service provider

4.4 Identifying exclusions

Depending on the situation, for the ISO 9001 requirements that are not applicable, it is possible to claim and document the exclusions. It is important to carefully read the requirements, identify and justify the possible exclusions. In our case, we identified three exclusions given in Tab. 2.

With all of these considerations, it is fairly straightforward to create the required documents based on the templates we obtained from one of the ISO 9001 consulting firms [29].

4.5 Mapping Scrum artefacts into ISO 9001 requirements

Scrum process assumes well defined team roles and process artefacts. When implementing QMS, we ask ourselves if there is a possibility to map between Scrum artefacts and ISO 9001:2008 requirements. The short answer to this question is yes, but certain things need to be considered. Scrum artefacts include the Product Backlog (i.e. in a form of User Stories), Sprint Backlog (Tasks), Burn-down charts and task boards, etc. Please refer to the Scrum learning and reference material available at [32]. Two additional reference documents that should be created in any software development enterprise using Scrum are the following:

Definition of done. This is typically an internal document providing guidelines on how to verify that the development team is delivering features that are truly done, not only in terms of functionality, but also in terms of quality as well. The definition of done is typically written in a form of a checklist of valuable activities required by the team members to follow in order to produce the desired quality of the software product. The definition of done can be specified and defined for each project separately as different project types may have different level of quality requirements. The definition of done has become a common artefact in modern and experienced Scrum teams [33].

Definition of ready. The requirements in Scrum are typically captured using User Stories that describe the desired features. Before jumping into the implementation and moving an item from the Product to the Sprint Backlog, the team has to agree that the requirement specification is clear, feasible, and testable. In other

Table 2 Documenting applicable ISO 9001 exclusions

Exclusion	Justification
7.5.2 Validation of processes for production and service provision	The company does not have any processes where deficiencies become apparent only after the product is in use
7.5.5 Preservation of product	The company does not have any service deliverable with limited shelf-life or where special handling techniques are needed to preserve the product
7.6 Control of monitoring and measuring equipment	The company does not use any equipment to monitor and measure processes for products delivered to the customer

Table 3 Mapping Scrum artefacts into ISO 9001 requirements

Requirement	Reference/Artefact(s)	Comment
Management Reviews (5.6.1)	Quality Manual	Agenda/minutes
Education, Training, Skills and Experience (6.2.2e)	Quality Manual	Scrum training certificates, Scrum materials, ISO 9000 materials (kit and videos)
Evidence that realization process and product meet requirements (7.1d)	Product backlog, Sprint backlog, Burn-down charts	Definition of done*
Requirements Review results (7.2.2)	Product backlog, Sprint backlog	Definition of done*
Design Inputs (7.3.2)	Product backlog	
Design Review results (7.3.4)	Product backlog, Sprint backlog	Definition of ready*
Design Verification results (7.3.5)	Sprint backlog, Burn-down charts	Definition of done*
Design Validation Results (7.3.6)	Sprint backlog, Burn-down charts	Definition of done*, Acceptance tests*
Design Change Reviews (7.3.7)	Product backlog, Sprint backlog	Definition of ready*
Validation of (Special) Processes(7.5.2d)	Exclusion	
Basis for Calibration (where not traceable to standards) (7.6 a)	Exclusion	
Calibration results (7.6)	Exclusion	
Validity of Previous Measurements when Equipment found out of Calibration (7.6)	Exclusion	
Audit results (8.2.2)	Quality Manual	Internal audit schedule, Internal audit checklist, Internal audit form
Evidence of Product Conformity, Acceptance Criteria and Release Authority (8.2.4)	Sprint backlog, Burn-down charts	Definition of done*, Acceptance tests*
Nature of Nonconformities, Subsequent Actions including Concession (8.3)	Sprint backlog, Burn-down charts	Documenting bugs or deficiencies as items in Sprint backlog; use of issue tracking tools
Results of Corrective Actions (8.5.2e)	Quality Manual, Sprint backlog, Burn-down charts	CAPA forms; successfully implemented using issue tracking tools*
Results of Preventive Actions (8.5.3d)	Quality Manual, Sprint backlog, Burn-down charts	CAPA forms; successfully implemented using issue tracking tools*

**Note: additional artefacts that need to be archived*

words, for each project, the Scrum team should have a clear definition of which features (i.e. user stories) are ready to be implemented. Typically, the whole team collaboratively has to agree that the requirement is clear. Then, an item is testable if there is an effective way to determine if the functionality works as expected. Typically this is provided in a form of acceptance tests.

Acceptance tests. The acceptance test specifications can be a useful artefact needed to conform to ISO 9001 requirements. Typically, there is 5-10 acceptance tests per user story, but they vary depending on the specifics of the requirements. Finally, an item in the Product backlog needs to be feasible, which means that its tasks can be completed within one sprint time frame.

CAPA forms. Additional consideration here is the selection of supporting software tools used to manage the Scrum process and maintain the required artefact documents. An interesting example is the use of issue tracking software aimed at tracking bugs and issues. In addition to this primary function, it can very successfully be used to manage and document corrective and preventive actions (CAPA) as required by ISO 9001. CAPA issues can be documented using paper forms, but they can be handled much more efficiently if we use an issue tracking software such as Bugzilla [33].

The mapping of the quality management system requirements to Scrum artefacts as it was implemented in this case study is given in Table III. Please note the addition of Definition of Done, Definition of Ready, and understanding of the software tools used to manage the Scrum process and QMS. The illustrated approach worked fine in a small enterprise environment and led to a successful ISO 9001:2008 certification.

5 Discussion

The development setup as described in this paper has been successfully used in a small software enterprise for two types of activities: a) migration of the C/C++ code base into Java; and b) development of the new projects from scratch. Migrating from C/C++ to Java turned out to be straightforward due to the similarities between the languages. Another aspect was the availability of various OSS libraries for Java that helped to avoid dependency on third-party commercial libraries. Various commercial libraries for C++ have been successfully replaced by using the OSS libraries available for Java (handling of images, archives, PDF, etc.). As for the development of the new projects, the described solution worked very well.

One of the things we got asked a lot is how we reconciled "document-intensive process" required by ISO 9001, which is not conformant to the principles of agile development. A question like this is wrong on both basis: a) ISO 9001 is not as document-intensive as it once was; and b) agile development process does not mean that there is no need for documentation at all. ISO 9001:2008 is really focused on establishing a meaningful QMS, which will, if implemented correctly, allow for continuous maintenance and improvement of the quality of services offered by the company. As far as documentation for Scrum and other agile processes, it can be minimized, but it is necessary. It boils down to making sure that we keep the history from the product backlog along with the reports of the user acceptance (Table III). This is very feasible, especially as most software companies do use software tools for Scrum project management such as RallyDev and ScrumDo [34, 35]. It is important to note that the ISO 9001 required documents (quality manual, policies and procedures) do not require frequent changes especially in small companies. Thus their maintenance is not time consuming.

The biggest challenge, as far as documentation goes, is handling of corrective and preventive (CAPA) issues, but, as discussed in precious section, they can be handled and tracked using issue tracking software. Each team may need to customize their CAPA process to fit their needs. Talking about software development services, there is a question about inevitable software bugs and their handling. Issue tracking software in combination with product backlog management tools can be used to treat software defects as newly discovered requirements. As such, work related to fixing software bugs can be prioritized, estimated, and planned for each iteration step.

6 Conclusions

The main contributions of the paper are summarized as follows:

- This paper describes agile software development for Java platform using Scrum and OSS tools. The paper demystifies ISO 9001 requirements and presents a case study for the implementation of QMS in a small software enterprise environment.
- The article describes a selection of OSS development tools for Java platform, which are put into the context of Scrum process and QMS implementation. The use of selected tools is illustrated with a deployment diagram describing the development setup.
- This paper describes the experience with QMS implementation and provides additional insight on challenging steps such as defining the company's organizational chart, process diagram, identifying and documenting exclusions, and mapping of the Scrum artefacts into ISO 9001 requirements.
- Finally, this case study can be used as a model for Scrum adoption and QMS implementation in small and medium software development enterprises, even if they use different programming languages. Being ISO 9001 certified may improve company's visibility and eligibility to bid for work with new clients.

Acknowledgements

The author would like to acknowledge the help and insights from the following individuals:

- Dr. Mladen Kezunovic with XpertPower™ Associates and Texas A&M University,
- Mr. Scott Dawson with Core Business Solutions.

7 References

[1] O'Reilly, T. Lessons from Open-Source Software Development. // Communications of the ACM. 32, 44(1999), pp. 33-37. DOI: 10.1145/299157.299164
[2] Open Source Initiative (OSI). URL: http://www.opensource.
[3] Cohn, M. Succeeding with Agile: Software Development Using Scrum (1st edn). Addison-Wesley Professional, 2009.
[4] Royce, W. Managing the Development of Large Software Systems, ICSE'87, 9th International Conf. on Software Engineering, 1987, pp 328-338.
[5] Larman, C. Agile and Iterative Development: A Manager's Guide. Addison-Wesley, 2004.
[6] Schwaber, K.;Beedle, M. Agile Software Development with Scrum, Prentice Hall, 2001.
[7] ISO 9001:2008. Quality Management Systems – Requirements. ISO Std., 2008.
[8] Williams, L.; Cockburn, A. Agile Software Development: It's about Feedback and Change. // IEEE Computer, June 2003, pp. 39-43.
[9] Vriens, C. Certifying for CMM Level 2 and ISO9000 with XP@Scrum. // Conference on Agile Development ADC'03, 2003.
[10] Stålhane, T.; Hanssen, G. K. The application of ISO 9001 to agile software development. // 9th International Conference PROFES 2008, Monte Porzio Catone, Italy, June 23-25, 2008. DOI: 10.1007/978-3-540-69566-0_30
[11] Wright, G. Achieving ISO 9001 Certification for an XP Company, in the book XP/Agile Universe 2003, LNCS 2753, Springer-Verlag Berlin Heidelberg 2003.
[12] 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
[13] 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.
[14] McMichael, M.; Lombardi, M. ISO 9001 and Agile development. // Agile 2007 Conference, 2007.
[15] Palmes, P. C. Process Driven Comprehensive Auditing: A new Way to Conduct ISO 9001:2008 Internal Audits, ASQ Quality Press, 2009.
[16] Biró, M. The Software Process Improvement Hype Cycle. // CEPIS UPGRADE Journal. 10, 5(2009), pp. 14-20.
[17] Qasaimeh, M.; Abran A. An Audit Model for ISO 9001 Traceability Requirements in Agile-XP Environments. // Journal of Software. 8, 7(2013), pp. 1556-1567. DOI: 10.4304/jsw.8.7.1556-1567
[18] Qasaimeh, M.; Abran A. Extending Extreme Programming User Stories to Meet ISO 9001 Formality Requirements. // Journal of Software Engineering and Applications. 4(2011), pp. 626-638. DOI: 10.4236/jsea.2011.411074
[19] Qasaimeh, M.; Abran A. Investigation of the Capability of XP to Support the Requirements of ISO 9001 Software Process Certification. // 2010 8th ACIS International Conference on Software Engineering Research, Management and Applications, SERA 2010, Montreal, Canada, May 2010.
[20] Shen, B.; Ruan, T. A Case Study of Software Process Improvement in a Chinese Small Company. // 2008 International Conference on Computer Science and Software Engineering, 2008. DOI: 10.1109/CSSE.2008.701
[21] Jakobsen, C. R.; Sutherland, J. Scrum and CMMI – Going from Good to Great: Are you ready-ready to be done-done? // Agile 2009 Conference, Chicago, 2009.
[22] Schiel, J. Implementing ISO 9001 with Scrum. Presentation Slides, 2008, URL: http://www.scrumalliance.com/resource_download/440
[23] Agile Manifesto. URL: http://www.agilemanifesto.org
[24] Schwaber, K. Agile Project Management with Scrum (1st ed). Microsoft Press, 2004.
[25] Cohn, M. User Stories Applied (1st edn). Addison-Wesley, 2004.
[26] Hoyle, D. ISO 9000 Quality Systems Handbook (6th edn). Taylor & Francis, 2009.
[27] Robbins, J. E. Adopting Open Source Software Engineering (OSSE) Practices by Adopting OSSE Tools. // ISCE'02, Orlando, FL, USA, 2002.

[28] Popovic, T. Agile Practices and Open Source Software Tools for Java Programming. // IT'13 Conf. Zabljak, Montenegro, 2013.
[29] Ubuntu Linux. URL: http://ubuntu.com
[30] 9000world. URL: http://www.9000world.com
[31] Quality Advantage. URL: ttp://www.qualityadvantage.com
[32] Scrum Alliance. URL: http://www.scrumalliance.org
[33] Pichier, R. Agile Product Management with Scrum: Creating Products that Customers Love (1st edn). Addison-Wesley Professional, 2010.
[34] Bugzilla. URL: http://bugzilla.org
[35] RallyDev. URL: http://rallydev.com
[36] ScrumDo. URL: http://www.scrumdo.com

Author's address

Tomo Popović, PhD
University of Montenegro
Faculty of Electrical Engineering
Dz. Vasingtona bb, 81000 Podgorica, Montenegro
E-mail: tpopovic@ac.me