Model-Driven Gap Analysis for the Fulfilment of Quality Standards in Software Development Processes

Giovanni Giachetti (giovanni.giachetti@unab.cl)
Universidad Andres Bello
José Luis de la Vara
Universidad de Castilla-La Mancha
Beatriz Marín
Universitat Politècnica de València

Research Article

Keywords: Quality standards, model-driven quality assurance, gap analysis, process configu-ration, ISO 9001, quality knowledge

Posted Date: January 27th, 2023

DOI: https://doi.org/10.21203/rs.3.rs-2329678/v1

Version of Record: A version of this preprint was published at Software Quality Journal on September 27th, 2023. See the published version at https://doi.org/10.1007/s11219-023-09649-x.

Abstract

Nowadays, society is moving to rapid digitalization of almost any aspect of our lives. From healthcare and transport to work or entertainment, we depend on software products that must function properly. The adequate construction of these software products can be evaluated and certified by means of quality standards. However, the adoption of quality standards is a complex and time-consuming task that requires experienced practitioners with advanced knowledge about the standards and development practices. This presents a dichotomy for development companies that are moving to more agile schemes, where the necessary expert knowledge about quality is not easy to get. To overcome these issues, this paper presents a model-driven approach to automate the analysis of software development processes that must be aligned to quality standards. The approach has been implemented in an open-source tool and applied to a case study to automatically generate a gap analysis by reusing expert knowledge related to the ISO 9001 standard. The application has demonstrated that is possible to reduce the effort required to perform the analysis and the improvement of development processes to assure the fulfilment of specific quality standards.

1 Introduction

Our society is rapidly moving to a highly digitalized ecosystem of existing and emerging technologies that affect almost every aspect of our lives(Fukuyama, 2018). Healthcare services, transport, working environments, or entertainment are currently controlled or intervened by interrelated systems in multiple manners(Deguchi et al., 2020). This presents important challenges for software developing companies that must satisfy a variety of users' needs while producing products that must function properly to prevent any damage to the people or economic loss(I. Lee et al., 2011; Leveson & Weiss, 2009). The adequate construction of software products can be evaluated and certified by means of quality standards (Rodríguez & Piattini, 2012). In this context, software-related companies that decide to adopt quality standards for their development processes require specific and specialized resources (human and technical ones) to properly configure the processes (Ayyagari & Atoum, 2019; Dahar & Roudies, 2018). Quality standards such as ISO 9001(Organization, 2015) or CMMi (Silva et al., 2015) are specified in textual documents, which must be interpreted by practitioners based on their experience. Since this is primarily a handmade and subjective task, which is highly dependent on expert knowledge (M.-C. Lee & Chang, 2006), it is difficult to provide automatisms for quality assessment of development processes according to organizational needs or specific domains (García-Borgoñon, Barcelona, García-García, Alba, & Escalona, 2014).

Furthermore, the current software development context is demanding faster and more agile development cycles to produce software products in almost any domain for a highly digitalized society (Digital.ai, 2021; Hallstedt, Isaksson, & Öhrwall Rönnbäck, 2020). In this context, lack of tools to facilitate the quality assurance of development processes can produce a negative effect when the time to market of quality product needs to be reduced (Ozkaya, 2021). The problem turns more complex when the knowledge needed to assure the fulfillment of certain quality standards is only in the head of a few experts (Edison, Wang, & Conboy, 2021; Qumer & Henderson-Sellers, 2008).

Therefore, three relevant issues can be observed. First, the quality assessment of development processes demands important time from specialized resources (Ayyagari & Atoum, 2019). Second, the knowledge about development practices that meet specific quality criteria is poorly documented or not documented at all(Qumer & Henderson-Sellers, 2008). And third, process configuration is slow and susceptible to present inconsistencies due to the lack of automatic or assisted support in relation to reference quality standards(Campanelli & Parreiras, 2015; Silva et al., 2015).

These issues can be considered as barriers for the companies that are trying to move to more agile development schemes and, at the same time, try to certify the quality of their software products in a society that is highly demanding quality digital goods. These barriers can be even higher for medium and small companies that cannot afford large and expensive certification processes (Basri & O'Connor, 2010; Machado, Mexas, Meza, & de Oliveira, 2022). To tackle these issues, it would be necessary to provide mechanisms for the representation of quality standards and reuse the expert knowledge related to the development practices that are necessary to meet different quality criteria. In addition, supporting tools for the verification of development processes are necessary to facilitate the identification of the gaps that must be bridged to fulfill the reference standards (de la Vara, Ruiz, & Blondelle, 2021).

As a solution, this article presents a model-driven approach for representing quality models that enables the automatic assessment of software development processes. The approach aims to reduce the effort involved, enable the reuse of expert knowledge about quality, and prevent errors generated from manual alignment of software development processes to specific standards.

This work has been validated by means of a case study that considers the update of an industrial development process with agile practices that must comply with the ISO 9001 standard. To perform this validation, a suite of model-based tools has been implemented. Thus, the contribution of this paper is twofold: 1) to introduce the model-driven approach used for representation of quality models and quality assessment, and 2) to show how the approach has been implemented to perform an automatic gap analysis of a development process according to a specific quality model.

The rest of the paper is organized as follows. Section 2 discusses the related work and the contribution of the approach beyond the state of the art. Section 3 introduces the conceptual foundation for the model-driven approach that supports the quality standards representation and development processes assessment. Section 4 presents the implementation of the approach in a suite of tools to perform automatic gap analyses of development processes. Section 5 analyses the approach by considering the results obtained from the industrial case study. Finally, Section 6 presents our main conclusions and future work.

2 Background, Related Work, And Contributions

Quality assessment of development processes requires experienced practitioners to evaluate and tailor different practices for the fulfilment of the quality criteria of one or more reference standards (García-Borgoñon et al., 2014). We analyse below prior work on quality model assessment and supporting tools, paying especial attention to model-driven proposals. Afterwards, we explain the main differences and contribution of the approach presented in relation to the state of the art.

Quality Standards and Development Process Assessment

There exist several quality standards related to software process improvement approaches (Pino, García, & Piattini, 2008; Unterkalmsteiner et al., 2011). The purpose of these standards is to increase the quality of final software products. Moreover, the development strategies are evolving to more agile processes, which demands that the quality assessment of these process is also aligned with this evolution (Poth, Sasabe, Mas, & Mesquida, 2019). Most of the quality assurance and certification approaches for agile methods involve traditional standards such as Capability Maturity Model Integration (CMMi) (Silva et al., 2015) or ISO 9001(Edison et al., 2021), which are not specific for agile development methods. From these standards, CMMi is the most adopted worldwide (Henriquez, Calvo-Manzano, Moreno, & San Feliu, 2022). There is a specific subset of CMMi for development process, which is called CMMI-DEV, that has been used as reference for other quality models such as MPT.Br (Furtado, Gomes, Andrade, & de Farias Junior, 2012).

For certification against quality standards, large and costly process are normally required. This hinders its adoption for all kind of development companies, especially for SMEs (Basri & O'Connor, 2010; Machado et al., 2022; Pino et al., 2008).

To better understand the main tasks necessary to achieve a quality certification, Fig. 1 shows a simplified quality assessment process, which focuses on the main elements involved in our work. The process considers, as an initial step (Step 1), the participation of an expert (internal or external to the organization) who captures the existing information about the organizational development processes (user manuals, guidelines, project configuration, development artifacts, tools, practices, records, etc.). The expert analyzes the information captured by considering the target quality standard (e.g., ISO 9001) to identify those quality criteria that are met (Step 2). This analysis is based on the expert's experience and knowledge. A gap analysis report is usually generated from the analysis performed, which shows the quality criteria met and those missing elements (gaps) that need to address to guarantee alignment with the target standard (Step 3). Once the gaps identified are bridged, the corresponding quality certification can be obtained (Step 4). The expert can also recommend alternatives to bridge the identified gaps. These recommendations depend on the experience of the consultant involved in the quality assessment process.

The certification process is complex, and the time required to do it varies depending on the expert knowledge, as well as the details about the configuration of development processes, and whether the decision made in the past are poorly or not documented (Edison et al., 2021; Kurapati, Manyam, & Petersen, 2012; Qumer & Henderson-Sellers, 2008). Even in those organization that properly document their development processes, there is always an important effort related to understand, configure and determinate the alignment of the development practices to the quality criteria to be met (Dahar & Roudies, 2018). This is related to the manner that the standards and development practices are documented, normally using textual specifications.

Moreover, the evaluation of the development processes of a company and its alignment with a specific standard is most often a manual task that can be partially supported by documentation traceability tools (Castellanos Ardila, Gallina, & Ul Muram, 2022). The results obtained are subjective and imprecise to be evaluated by third parties because they depend on an expert's experience to perform the gap analysis against the reference quality model.

Model-Driven Tools for Management of Quality Standards

Model-driven approaches are based on conceptual models to provide a formal description of world-related elements. These conceptual models are intended to facilitate understanding and communication (Mylopoulos, 1992) among different actors, and to facilitate the implementation of tools by following a model-driven development (MDD) process (Pastor, España, Panach, & Aquino, 2008). For the specification of these conceptual models, graphical notations combined with textual information can be used (Engelen & van den Brand, 2010). This is particularly interesting for quality models, where existing standards are normally textual specifications despite the fact that there are studies that demonstrate that graphical representations are preferred over textual models and can facilitate, e.g., understanding (de la Vara, Marín, Ayora, & Giachetti, 2020; Weske, 2007). Thus, conceptual modeling of quality standards can be regarded as the Rosetta stone to follow a model-driven approach for quality assessment of development process.

Some approaches have proposed specific mappings to indicate how agile practices can fulfill quality criteria of certain standards. This is the case of the approaches related to CMMi, ISO 9001, and their combinations with other quality models (Dahar & Roudies, 2018; Ijaz, Asghar, & Ahsan, 2016; M.-C. Lee & Chang, 2006; Walz & Caroll, 2011) to evaluate software and product quality. Counting on formal mechanisms to represent this information is useful to implement quality assessment supporting tools and to facilitate the interchange of knowledge among different domains and adoptions of quality standards. In this context, there are model-driven approaches that remark the relevance of expert-knowledge management for the proper configuration of agile development processes and quality assessment (Qumer & Henderson-Sellers, 2008; Singh, Singh, & Sharma, 2012).

Main Contribution Beyond the State of the Art

As mentioned above, quality assessment of a development process demands expert knowledge to determine if the relevant quality criteria are met by one or more development practices involved in the process. Thus, the first contribution presented in this work is to propose a conceptual representation of quality standards to be met, which will be supported by model-driven tools to automate the analysis of the standards represented.

Furthermore, the complexity of a quality assessment is also related to the number of possibilities for the configuration of a development process and the number of different ways that a quality criterion can be met. A quality standard can consider a huge amount of quality criteria to be evaluated. As mentioned above, the alignment of quality criteria with one or more development practices is normally in the head of a few experts. Thus, a second contribution of our approach is to support the specification of the relationships between the development practices and the quality criteria defined. With this information, it will be possible to reuse the experts' knowledge and to automate the identification of gaps to be bridged according to the reference quality standards. Moreover, this would enable that the expert knowledge is reused in different projects and even that it is interchanged across different organizations.

Figure 2 shows how the quality assessment and certification process can be improved with the contributions proposed. In this improved process, the first step is assisted by modelling tools that facilitate process definition and verification, thus reducing the effort and time required. The second and third steps of the process can be totally automated to generate a model-driven gap analysis report. Finally, a set of recommendations based on the reuse of expert knowledge will be automatically generated to improve the process towards quality certification.

Finally, it is important to note that, in the improved process presented in Fig. 2, the participation of an expert in the target quality standard will not be necessary. The approach proposed will allow less experienced practitioners to perform the quality assessment and configuration of development processes in a similar way to how an experienced practitioner can do it. The next section introduces the model-driven approach defined to obtain the contribution mentioned.

3 Modeling Quality Standards And Quality Assessment

Two elements are needed to perform the quality assessment of development processes by following a model-driven approach: from one side, the proper specification of the reference quality standard; from the other side, the provision of mechanisms to link the quality practices defined in the quality models with the process defined to assess if a process activity and its results fulfill the corresponding quality criteria.

3.1 Modeling Quality Standards

For the representation of quality standards, a conceptual representation following a model-driven approach has been defined. This means that the main concepts, properties, and relationships are described by using a metamodel. This metamodel can be later instantiated to represent different quality standards specifications. Two well-known standards have been considered: CMMi and ISO9001. These standards are considered because they are widely adopted in software projects (Mutafelija & Stromberg, 2003) and the companies involved in the proposal are certified under these standards.

For conceptual representation, the CMMi-DEV version 1.3 (SEI, 2010) has been considered. CMMI-DEV proposes 22 *process areas*, which are related to different stages of a software development process, such as *project planning*, *requirement management*, *verification*, *validation*, etc. Each process area considers a set of goals to be achieved, which can be specific or generic goals. Specific goals are related to one process area only. For instance, *to develop a project plan* is a specific goal for the *Project Planning* (PP) process area. Generic goals are related to all process areas, for instance, *the process is institutionalized as a quantitatively managed process*. Moreover, a set of *Practices* must be implemented to achieve the corresponding goals. These practices can be of two types, specific or generic, accordingly to the type of goal involved. Thus, the main elements involved in the CMMI model are *Process area*, *Goal* (generic or specific), and *Practice* (generic or specific).

Furthermore, the CMMi model considers different levels depending on how it is used. It can be used in a continuous manner that is centred on capability levels, or in a staged manner that is focused on maturity levels. Each level builds on the previous level, *i.e.*, Level 2 considers Level 1 fulfillment together with those practices needed to achieve the goals of Level 2; Level 3 considers Level 2 fulfillment together with those practices needed achieve the goals of Level 3, and so on till Level 5.

The approach proposed in this paper considers the staged use of CMMi, which is based on five maturity levels.

- **Level One (initial)** considers that the development processes are chaotic, which means that these are poorly or not controlled. Success in these organizations depend on the competence and heroics of the people in the organization and not on the use of proven processes.
- **Level Two (managed)** involves project management and support practices to systematically translate software requirements to quality-accepted products.
- **Level Three (defined)** involves systematic processes properly characterized in standards and procedures.
- **Level Four (quantitatively managed)** involves that processes are continually measured to assure the quality of the resultant software products.
- **Level Five (optimizing)** involves the continuous improvement of processes trough organizational innovation.

The relationships between the CMMi concepts, when the model is applied in a stagged manner, can be summarized as follows: the process areas belong to a maturity level and must fulfill specific and generic goals. For achievement of these goals, specific and/or generic practices must be implemented (see Fig. 3).

The other quality standard considered is ISO 9001, which is widely known and adopted for defining a management system oriented to product and service quality by meeting customer requirements (Dahar & Roudies, 2018). In particular, the ISO 9001 version 2015 (Organization, 2015) has been considered as reference. ISO standards are based on the Plan, Do, Check, Act (PDCA) cycle to achieve continuous improvement (da Fonseca, Domingues, Machado, & Harder, 2019). The ISO 9001 standard is organized in 10 clauses related to the different PDCA phases.

- The **Plan** phase is related to the the *Scope*, *Normative references*, *Terms and Definitions*, *Context of the organization*, *Leadership*, and *Planning* clauses.
- The **Do** phase is related to the *Support* and *Operation* clauses.
- The **Check** phase is related to the *Performance evaluation* clause.
- The **Act** phase is related to the *Improvement* clause.

The main clauses are detailed in 56 subclauses related to the different aspects that must be considered to obtain a proper organizational quality management system. These subclauses are finally related to specific requirements that indicate specific practices that must be implemented by the company. Once the different requirements related to a clause and its corresponding subclauses are met, then the company complies with the quality standard in that clause. Figure 4 shows the organization of the main concepts related to ISO 9001 by means of an example based on the subclause 8.2.2 *Determining the requirements for products and services*.

From Fig. 4 it can be observed that the main concepts related to the ISO 9001 standard are *Clauses*, *Subclauses*, and *Requirements*. These concepts have a structural organization like the CMMi concepts related to *ProcessArea*, *Specific Goals*, and *Specific Practices* (see Fig. 5). Thus, it would be possible to use these conceptual equivalences in the definition of a metamodel that supports the definition of quality models for the automatic gap analysis of development processes considered in our approach.

The adoption of both ISO 9001 and CMMI standards has already been studied to generate software process improvement (SPI) schemes (Ijaz et al., 2016; M.-C. Lee & Chang, 2006). It is also interesting to observe that there are approaches that combine these reference standards with other quality models, such as MPS.BR (Ferreira et al., 2007; Montoni, Rocha, & Weber, 2009) or IEEE std 730(Walz & Caroll, 2011), finding correspondences among them to obtain novel approaches for the improvement of software processes and of the final software products (Ferreira et al., 2007; Walz & Caroll, 2011). This would facilitate the application of our approach to different development contexts. Furthermore, with the high adoption of agile development methods in the last years, there have emerged approaches that combine quality models to generate assessment frameworks that can are customized according to specific agile development practices (Pardo-Calvache, Chilito-Gómez, Viveros-Meneses, & Pino, 2019).

A specific metamodel for quality standards has been defined (see Fig. 6) to support different instances of quality models with a common meta-specification. Despite this metamodel has been designed to generate instances of CMMi and ISO 9001 models, it also considers improvements that can be useful to represent other standards, such as recursive relationships in different constructs and flexibility in the use of quality levels.

The structure of the metamodel for quality standards considers that each model instantiated is representing a specific standard. The process areas are instances of the metaclass *ProcessArea*. To provide flexibility to the semantic of this construct, a process area can have subprocess areas, represented with the recursive relationship *hasSubProcess*. The process areas can be related to none or many levels; thus, it is possible to support standards that have different certification levels such as CMMi.

A process area can have different goals related (metaclass *Goal*) and the goals can have many practices (metaclass *Practice*). Both constructs, goals and practices, have also recursive relationships to provide additional representation features in case a standard requires the definition of sub-goals or sub-practices, represented by the relationships *hasSubGoal* and *hasSubPractice*, respectively.

Moreover, goals and practices have the attribute *type* to indicate the generic or specific property according to the CMMi specification. These types are represented by the enumerations *GoalType* and *PracticeType*. Although the value of the type is the same in both cases, two enumerations have been defined to facilitate the customization of the metamodel in case it is used for other standard with different values for these types and conceptual constructs.

The metaclass *Model* also includes a property *prefix* to refer to the quality standard represented This facilitates the identification of quality elements when different quality standards are applied over the same process. From the other side, the property *isValidated* will be used to indicate if the model defined has already been validated, for instance, by auditors or organizational experts. This will be also useful to manage different versions of quality standards, or when there are non-standardized quality models for internal use, such as self-defined organizational quality practices.

3.2 Linking Quality Models and Development Processes

A specific weaving metamodel is defined to represent the links that may exist between the quality models and the different development components that will be used in the configuration of specific development processes. Figure 7 shows a simplified schema of the inputs for the model-driven quality assessment that involves the quality models of the reference standards and for the weaving from the quality practices and work products to the different development components that are expressed in terms of development methods and artifacts. These development components are used to define specific development process models.

The quality weaving models are defined by considering expert knowledge to indicate those development practices that enable the fulfilment of specific quality practices. *i.e*, expert knowledge related to the alignment of development processes and quality models is expressed by means of these weaving models. The weaved development components are stored in a repository that will be used to define concrete development processes according to project needs.

The quality weaving models are described by means of a metamodel (see Fig. 8) that consists of three main constructs, namely *Activities*, *Artifacts*, and *Quality Practices*, represented by the metaclasses *ActivityRef*, *ArtifactRef*, *QualityPracticeRef.* All these constructs are part of the development method models, artifact models, and quality models, which are referenced by the metaclasses *MethodModelRef*, *ArtifactModelRef*, and *QualityModelRef*.

A specific link is defined in the weaving metamodel to connect the elements from the development perspective to quality practices by means of the metaclass *WorkProductLink*. This metaclass represents the effort necessary (activities) to obtain an outcome (work product) that fulfills specific quality aspects. This effort can be performed by following specific practices, thus connecting a quality practice from a quality model with activities and artifacts from development methods. The reference to a quality practice is mandatory for the definition of a work product link. This is represented with the cardinality one-to-one from *WorkProductLink* to *QualityPracticeLinkEnd*. One or more activities and artifacts are also necessary to define work product links. This is represented with the cardinality one-to-many from *WorkProductLink* to *ActivityLinkEnd* and *ArtifactLinkEnd*. A concrete example of the instantiation of the quality weaving model is presented in the next section.

Furthermore, the work product concept is a relevant element to perform the quality assessment over development processes to determine the fulfilment (or not) of the different quality criteria to be met. Further ontological analyses of the concept of work product can be found in (Gonzalez-Perez, Henderson-Sellers, McBride, Low, & Larrucea, 2016; Ruy, de Almeida Falbo, Barcellos, & Guizzardi, 2014) for the ISO/IEC 24744 standard, which demonstrate its relevance in quality assessment for development process. Moreover, the work product concept can be used to represent product-centred development approaches (Gonzalez-Perez & Henderson-Sellers, 2008), which are useful for configuration of agile processes.

The next section exemplifies the application of the approach by means of a suite of model-driven tools. These tools automate the generation of a gap analysis report to perform the quality assessment of a development process to fulfill specific quality standards.

4 Implementing The Model-driven Gap Analysis

A suite of Eclipse-based tools has been implemented to support the proposed approach for the definition of models of quality standards and quality weavings. The tools also support the specification of development processes that are automatically analyzed to obtain a gap analysis according to specific reference standards. The implementation of these tools is based on the Eclipse plug-ins Sirius, M2Doc, and Acceleo (EclipseFoundation).

Two main stages are necessary to put the approach into practice with the tools implemented (See Fig. 9). The first stage is related to the representation of the quality models and the quality weavings. Expert knowledge related to quality evaluation is used to create a repository with information of development components and their relationships to specific quality criteria. This repository, called *Development Repository*, is defined only once, and it can be reused to define, analyze, and refine development processes related to different projects. The second stage considers the configuration of development processes using the information from the development repository. This second stage is conducted by the responsible for the development process, who does not need to be an expert in the reference quality standards. Finally, an automatic quality assessment is performed form the process defined by using the different models considered in the approach (see Fig. 7). As a result, a gap analysis report is obtained. This report indicates the quality practices met and not met by the process as well as alternatives to adjust the defined process to fulfill the target quality standard.

The two stages proposed for implementing the model-driven gap analysis are exemplified below with the evaluation of a set of ISO 9001 practices over an agile development process.

Stage 1: Represent Expert Knowledge about Quality

Figure 10 shows an excerpt of the ISO 9001 model defined with the model editor implemented from the quality metamodel proposed (see Fig. 6). In this example of ISO 9001 model, two process areas are defined to represent the clauses 7 and 8. Clause 7 is related to the support process area and Clause 8 is related to the operation process area. Each clause has one or more sub-clauses, which are represented by means of the goal concept. Finally, five quality practices are presented in red squares. The ISO 9001 standard does not have quality levels as such and, hence, it is represented that this quality model has one level only.

Figure 11 shows the quality weaving defined for a set of agile development practices related to the pre-game phase of the Scrum method, which will be used to define an illustration of a development process. The weavings are defined as work products links according to the quality weaving metamodel presented in the previous section (Fig. 8). The quality weaving model presents five mappings, one for each practice of the quality model (see Fig. 10), with a wide green box. The quality practice is represented by means of a small green box. The artifacts are colored in light blue and the activities in orange. Each work product link indicates the set of artifacts and activities from the Scrum development method that provide the necessary evidence to fulfill a specific ISO 9001 practice. For instance, the example quality weaving model indicates the quality practice (Fig. 11 work product *8.1 A*) can be met with the back log definition and the product back log as evidence artifact of the achievement of this practice.

It is important to mention that although the example only shows one mapping for each quality practice, the quality weaving supports the definition of many mappings for a same quality practice. Thus, it is possible to manage different alternatives to configure a development process properly.

Stage 2: Gap Analysis of Development Processes

For the example, a development process based on Scrum is defined. This step is performed by the responsible for the development process, which is normally the development team leader. This person does not need to be an expert in the reference quality standards. Therefore, it is possible to take advantage of the expert knowledge for process definition even though the responsible may be less experienced in quality assessment of development methods.

Figure 12 shows the initial development process defined for the example. The process editor uses a notation based on BPMN(OMG, 2011). Activities and Artifacts are represented with a divided box. The upper part of the box indicates the name or the element, and the lower part indicates the development element referenced from the development repository. This representation allows to use different development activities from the repository to configure a task in the development process. For instance, the example shows that the task *RequirementSpecification* is performed according to the Scrum activity *DefineProductObjetives*. This is similar for the task *DefineScrumRoles*, where the development activities *DefineScrumMaster* and *DefineProductOwners* are considered. Thus, different practices from different development methods can be used for process tailoring, which is particularly useful in the configuration of agile development processes (Al-Zewairi, Biltawi, Etaiwi, & Shaout, 2017; Mahanti, 2006).

In the case of artifacts, it is possible to reference to one or more artifacts defined in the development repository. For instance, the artifact *GanttChart* is referencing *ReleasePlan* and *MeetingDates*, which are artifacts of the Scrum method. This reference indicates that the Gantt Chart obtained in the process contains the information of meeting dates and the release plan according to the Scrum definition.

The process defined is used as input for the model-driven gap analysis according to the ISO 9001 model defined. The analysis of the process initially evaluates its completeness in terms of input/output artifacts, input/output flows for each activity, start and end tasks, and the validation of each task, artifact, and their dependency in relation to the reference development methods. Any issue found in the process must be solved to perform the analysis in relation to the quality model. In the quality assessment, the different quality criteria of the reference model and quality levels (if applicable) are analyzed by using the quality weavings defined. This analysis automatically identifies if there exist a set of tasks and artifacts that meet the related quality practices, generating a gap analysis report as the one shown in Table 1 for the example agile process. The report also indicates the alternatives that can be used to be fill the gaps identified.

Table 1. Results of the gap analysis performed to the example development process.

Process Area	Quality Practice	Method Activities Required
CLAUSE 7 SUPPORT	7.1.6 - The organization must determine the knowledge necessary for the operation of its processes and to achieve conformity of products and services **State:** Not Fulfilled	"Scrum Method": (Not Included) **DefineProductVision** (Included) **DefineScrumTeam** Output artifacts: (Included) **VisionDoc** (Included) **ScrumTeamRoles**
CLAUSE 8 OPERATION	8.3.3 A - The organization must consider functional and performance requirements. **State:** Not Fulfilled	"Scrum Method": (Included) **DefineProductObjectives** Output artifacts: (Included) **ProductRequirements** (Not Included) **ProductBacklog**
CLAUSE 8 OPERATION	8.3.3 B - The organization must consider information from previous similar design and development activities **State:** Not Fulfilled	"Scrum Method": (Included) **DefineBacklog** (Included) **DefineProductOwners** Output artifacts: (Not Included) **ProductBacklog**
CLAUSE 8 OPERATION	8.1 A - Define requirements for products and services **State:** Not Fulfilled	"Scrum Method": (Included) **DefineBacklog** Output artifacts: (Not Included) **ProductBacklog**
CLAUSE 8 OPERATION	8.1 B - Indicate the resources necessary to achieve the requirements of products and services. **State:** Not Fulfilled	"Scrum Method": (Included) **DefineBacklog** Output artifacts: (Not Included) **ProductBacklog**

The results of the gap analysis indicates that the gaps identified can be solved by adding the scrum activity *DefineProductVision* and the output artifact *ProductBacklog*. The refined process presented in Fig. 13 is obtained after addressing the gaps identified. This process fulfills all the quality criteria of the example ISO 9001 model.

5 Application Of The Automatic Gap Analysis Approach

This section summarizes the results obtained from applying the model-driven gap analysis approach to an industrial development process that has been updated to include agile development practices. From this industrial process, an exploratory case study following the guidelines provided by (Runeson, Host, Rainer, & Regnell, 2012; Wohlin et al., 2012) was performed considering the following research questions:

RQ1

Does the model-driven gap analysis proposed enable the reuse of expert knowledge to define development processes aligned with specific quality standards?

RQ2

Can the use of the model-driven gap analysis proposed result in a reduction of the effort needed for the definition of development processes correctly aligned with specific quality standards?

Case

The case corresponds to a development process applied in industrial projects for more than 10 years for a large multinational software development company that counts with more than 1.400 employees in four different countries

This process is based on an open-reference called Tutelkan (Valdés, Visconti, & Astudillo, 2011) and complies with CMMi-DEV (v1.2) (Team, 2006) and ISO 9001 (ISO 9001:2000) (Mutafelija & Stromberg, 2003; Organization, 2000).

The original development process specification corresponds to a large document (169 pages) that describes the activities, artifacts, and practices that can be configured for developing specific projects. This specification is textual only. The purpose is to use the model-driven approach proposed to specify a model for the updated development process, in order to obtain a gap analysis report that guides the process configuration to fulfill with the updated ISO 9001:20015 version. The model defined is centered on the Clauses 7 to 10 from the 10 Clauses of ISO9001:2015, which are the relevant ones at the company for the new certification of their development process.

Furthermore, only a subset of the complete development process specification is normally used in the configuration of development projects. This is since the complete process considers different development practices that must be tailored according to the specific needs of each project. The time required to manually configure this subset of activities involved in a specific development project is usually around two or three weeks. This process configuration requires the participation of a process analysis expert and a quality assurance (QA) expert to validate that the process defined fulfills the different organizational quality criteria.

Subjects

A group of nine industry experts participated in the specification of the quality model and quality weavings related to the clauses 7 to 9 of ISO 9001:2015. All the experts have more than 10 years of experience in project management, adoption of agile development methods, and/or quality certification processes. Some of the industry domains that they have worked on are banking, retail, telco, and civil aviation. Table 2 indicates each expert's specific knowledge. It can be observed that experts related to the CMMi standard also participated in development repository definition. These experts provided knowledge to maintain the consistency between the updated process and CMMi-DEV for future certification under this standard[1].

Table 2
Summary of the experts' knowledge.

Expert	Project Management	Agile Consulting	ISO 9001	CMMi-DEV
Expert 1	X		X	
Expert 2	X			
Expert 3	X			
Expert 4		X		
Expert 5	X	X	X	
Expert 6	X	X		
Expert 7	X			X
Expert 8	X		X	
Expert 9	X			X

Execution of the case study

The first stage of the application of the approach considers the definition of quality models and quality weavings to generate the reference development repository. This definition was an iterative and incremental process that took 6 months. Weekly meetings of one hour were held to analyze the development elements and their relationship with the quality criteria, and to achieve consensus with the different experts. The specification and alignment of the development practices with the quality standards involved the analysis of different-size projects developed by the company. The specifications of these projects were in textual documents and excels sheets, and most of the design decisions taken were in the head of the experts only. Thus, additional effort was necessary to define the corresponding process models to understand and properly specify the development practices and quality weavings that finally comprise the development repository. The development practices considered the original development method of the organization together with practices from the user experience SCRUM (Schwaber & Beedle, 2002), and Extreme Programming (XP) (Jeffries, 2001) development methods. Finally, the obtained ISO 9001 model has 43 goals and subgoals, and 121 quality practices[2]. The quality weavings defined involved 264 different work products related to different development practices. Table 3 summarizes the modeling elements defined for the clause 7 to 10 of ISO 9001:2015. In this table it can be observed that Clause 8 is the one with the highest number of quality practices and work products for quality weavings. The reason is that Clause 8 is related to operation of the quality management systems and involves the aspects of requirements, design, and development of products, which are central elements in a development process.

Table 3
Summary of the modeling elements defined for the ISO 9001:2015 - Clauses 7 to 10.

Modeling Element	Clause 7	Clause 8	Clause 9	Clause 10
Goals and Subgoals	8	29	3	3
Quality Practices	11	84	10	4
Work Products	18	225	17	4

The second stage related to the model-driven gap analysis approach considers the specification of the model of the development process from the information of the development repository. This task was performed by a process analyst that did not participate in the development of the repository. This decision aimed to determine if the information provided by the gap analysis report is complete enough to guide the configuration of the process to fulfill the target quality standard.

The analyst defined an initial version of the process considering the different development components available in the repository. This initial process was verified by the tool to evaluate its completeness and to generate the initial gap analysis report that was used to define an improved version of the process. The analyst made several iterations in a trial-and-error process guided by the tool, performing new gap analyses and addressing the issues identified until all the gaps were solved.

With the application of the proposed approach, the complete development process (not a subset) was configured by the process analyst in five days only, without the participation of a QA expert. This was possible because all the quality issues were identified by the model-driven gap analysis, which also indicated the alternatives for addressing the gaps identified in the development process. The resultant process was evaluated by the nine experts that participated in the definition of the development repository. They considered that it was properly aligned to the ISO 9001:2015 standard. Finally, the updated process passed an external ISO 9001 certification in 2020.

Analysis of the results

The results obtained from the case study show that a process analyst that is not an expert in the evaluation of the ISO 9001 standard, was able to define a new and compliant version of the development process by following the information from the development repository. The process configured passed the quality experts' validation to determinate the alignment with the ISO 9001 standard, which also passed the external certification. Therefore, it is possible to answer the first research question *RQ1: Model-driven gap analysis proposed allows to reuse expert knowledge for defining development process aligned with quality standards?* positively.

Furthermore, the effort related to the definition the development process is reduced in relation to the manual process configuration without the support of the model-driven gap analysis; from two weeks for a subset of the original development process, to 5 days for configuring the complete development process. Moreover, it has especial relevance that the process configuration was performed without the support of a QA expert, thus also reducing the effort necessary from more experienced human resources. Considering this preliminary evidence, we can also have an affirmative answer for the second research question. *RQ2: The effort needed for the definition of development process correctly aligned with specific quality standards is reduced?*

However, this answer is affirmative only if the time related to the development repository is not considered. This and other issues from the evaluation performed are analyzed in the threats to validity section.

Threats to validity

Even though we have performed the study carefully following wide-known guidelines, there are some threats to the validity of our results that we discuss below.

Construct validity. It is not possible to effectively compare the time required to define the development repository in relation to the time required for the previous certification of the development process. Experts indicated that previous certification of the process took about one year (under the ISO 9001:2000 standard). Although the definition of the development repository took 6 months, including CMMi information, the previous experience of the experts generates a bias in the results. This previous experience can have a positive impact in reducing the time required to perform the matching between the development components and the different quality practices. The effort reduction obtained in the case study is a preliminary promising result. However, further empirical studies for its confirmation are necessary.

Internal Validity. The training time to apply the approach is not considered in the evaluation performed. This training time took around one week for the use of the process editor and the gap analysis tool by the process analyst. This time only applies the first time the tool is used, so it can be considered as learning time that will diminish over time. Moreover, once the analyst has experience with the tool, the time is not relevant for the configuration of new processes. However, it can be considered a bias in relation to the effective effort required in the case study.

External validity. The approach has been applied to a single company only, so it is not possible to generalize the results to any company that need to perform a gap analysis to fulfil quality standards. Nevertheless, it is representative for companies that are updating their development process, for instance, to adopt agile practices. We are aware that to improve the generalization of our results it is necessary to perform more studies on different projects and organizations, and to count on adequate data about the effort reduction that the approach provides for effective alignment to quality criteria of different standards.

[1] The CMMi-DEV model defined can be downloaded from https://zenodo.org/deposit/7378525.

[2] The ISO 9001 model defined can be downloaded from https://zenodo.org/deposit/7378525. The original model is in Spanish

6 Conclusions And Future Work

This has paper presented a model-driven approach to facilitate the quality assessment of development processes according to specific standards by of means a gap analysis that is automatically conducted. To support this automatic gap analysis, the approach relies on a conceptual framework to specify models of different quality standards and of their alignment to specific development practices. The alignment of quality models and development practices is based on expert knowledge about quality, which can be reused in the assessment of different development projects. This is of paramount importance in a software development context that is constantly changing and in which expert knowledge is scarce and volatile. Therefore, it takes special relevance to count on mechanisms to keep and take advantage of this knowledge.

Furthermore, the development companies are evolving to adopt more agile practices to deliver high-quality software products in less time, thus contributing to tackling novel challenges of a society whose digitalization is growing. In this digital society, the quality of the software development processes takes special relevance when the resultant software products are related to critical processes or systems. Hence, the quality assessment of development processes must not be a problem to improve the delivery of software products. It is necessary to count on quality assurance mechanisms that are aligned with faster software production processes, also reducing cost and effort, as we have demonstrated with the approach proposed in this paper.

Moreover, it is important to facilitate the access and interchange of expert knowledge about quality by means of mechanisms that democratize the access to quality certification processes that improves the development capabilities of different size companies. The conceptual approach proposed, and the supporting tools implemented have demonstrated in practice that it is possible to reuse expert knowledge about quality by less experienced practitioners. This brings another benefit, which is a reduction of the risk of losing valuable expert knowledge due to the mobility of the practitioners. As informative note, 50% of the professionals originally involved the case study presented have moved to other companies. However, it is still possible to use their expert knowledge through the information of the development repository defined.

Future research will consider the application of the approach to other quality standards, such as quality standards related to healthcare technologies and safety certification of critical systems. In addition, the preliminary results of the case study show that effort can be reduced when the approach is used to configure development processes for different projects. Thus, we are working on new empirical evaluations to measure the impact in effort reduction from using expert knowledge about quality among different organizations and projects.

Declarations

Funding. The work leading to this paper received funding from the iRel4.0 (H2020-ECSEL grant agreement no 876659; Spain's MICINN ref. PCI2020-112240), VALU3S (H2020-ECSEL grant agreement no 876852; Spain's MICINN ref. PCI2020-112001); Treasure (JCCM ref. SBPLY/19/180501/000270; EC's European Regional Development Fund) projects; The Ramon y Cajal Program (Spain's MICINN ref. RYC-2017-22836; EC's European Social Fund); and SIAM (Chile's CORFO-INNOVA ref. 16COTE-60218), and from NISUM Technologies Chile.

Authors contribution. Giovanni Giachetti conceived the proposal and wrote the main manuscript text. José Luis de la Vara performed the conceptual validation of the model-driven approach. Beatriz Marín supported the empirical validation of the approach. Giovanni Giachetti, José Luis de la Vara, and Beatriz Marín prepared and reviewed the manuscript.

Data availability. The diagrams of the modelling artifacts related to the proposal application have been deposited in Zenodo repository with the reference code 7378525.

Conflict of interest. The authors declare no competing interests.

References

1. Al-Zewairi, M., Biltawi, M., Etaiwi, W., & Shaout, A. (2017). Agile software development methodologies: survey of surveys. *Journal of Computer and Communications, 5*(05), 74.
2. Ayyagari, M. R., & Atoum, I. (2019). CMMI-DEV Implementation Simplified. *International Journal of Advanced Computer Science and Applications, 10*(4).
3. Basri, S., & O'Connor, R. V. (2010). *Understanding the perception of very small software companies towards the adoption of process standards.* Paper presented at the European Conference on

Software Process Improvement.

4. Campanelli, A. S., & Parreiras, F. S. (2015). Agile methods tailoring–A systematic literature review. *Journal of Systems and Software, 110*, 85-100.
5. Castellanos Ardila, J. P., Gallina, B., & Ul Muram, F. (2022). Compliance checking of software processes: A systematic literature review. *Journal of Software: Evolution and Process, 34*(5), e2440.
6. da Fonseca, L. M. C. M., Domingues, J. P., Machado, P. B., & Harder, D. (2019). ISO 9001: 2015 adoption: A multi-country empirical research. *Journal of Industrial Engineering and Management (JIEM), 12*(1), 27-50.
7. Dahar, H., & Roudies, O. (2018). *Measurement of Co-deployment of IT Quality Standard: Application to ISO9001, CMMI and ITIL.* Paper presented at the International Conference Europe Middle East & North Africa Information Systems and Technologies to Support Learning.
8. de la Vara, J. L., Marín, B., Ayora, C., & Giachetti, G. (2020). An empirical evaluation of the use of models to improve the understanding of safety compliance needs. *Information and Software Technology, 126*, 106351.
9. de la Vara, J. L., Ruiz, A., & Blondelle, G. (2021). Assurance and certification of cyber–physical systems: The AMASS open source ecosystem. *Journal of Systems and Software, 171*, 110812.
10. Deguchi, A., Hirai, C., Matsuoka, H., Nakano, T., Oshima, K., Tai, M., & Tani, S. (2020). What is society 5.0. *Society, 5*, 1-23.
11. Digital.ai. (2021). *15th Annual State of Agile Report*. Retrieved from https://stateofagile.com/
12. EclipseFoundation. Eclipse Modeling Tools. Retrieved from https://www.eclipse.org/downloads/packages/release/2022-09/r/eclipse-modeling-tools
13. Edison, H., Wang, X., & Conboy, K. (2021). Comparing Methods for Large-Scale Agile Software Development: A Systematic Literature Review. *IEEE Transactions on Software Engineering*.
14. Engelen, L., & van den Brand, M. (2010). Integrating textual and graphical modelling languages. *Electronic Notes in Theoretical Computer Science, 253*(7), 105-120.
15. Ferreira, A. I. F., Santos, G., Cerqueira, R., Montoni, M., Barreto, A., Barreto, A. O. S., & Rocha, A. R. (2007). *Applying ISO 9001: 2000, MPS. BR and CMMI to achieve software process maturity: BL informatica's pathway.* Paper presented at the 29th International Conference on Software Engineering (ICSE'07).
16. Fukuyama, M. (2018). Society 5.0: Aiming for a new human-centered society. *Japan Spotlight, 27*(5), 47-50.
17. Furtado, A. P. C. C., Gomes, M. A. W., Andrade, E. C., & de Farias Junior, I. H. (2012). *MPT. BR: A Brazilian maturity model for testing.* Paper presented at the 2012 12th International Conference on Quality Software.
18. García-Borgoñon, L., Barcelona, M. A., García-García, J. A., Alba, M., & Escalona, M. J. (2014). Software process modeling languages: A systematic literature review. *Information and Software Technology, 56*(2), 103-116.

19. Gonzalez-Perez, C., & Henderson-Sellers, B. (2008). A work product pool approach to methodology specification and enactment. *Journal of Systems and Software, 81*(8), 1288-1305.
20. Gonzalez-Perez, C., Henderson-Sellers, B., McBride, T., Low, G. C., & Larrucea, X. (2016). An Ontology for ISO software engineering standards: 2) Proof of concept and application. *Computer Standards & Interfaces, 48*, 112-123.
21. Hallstedt, S. I., Isaksson, O., & Öhrwall Rönnbäck, A. (2020). The need for new product development capabilities from digitalization, sustainability, and servitization trends. *Sustainability, 12*(23), 10222.
22. Henriquez, V., Calvo-Manzano, J. A., Moreno, A. M., & San Feliu, T. (2022). Agile-CMMI V2. 0 alignment: Bringing to light the agile artifacts pointed out by CMMI. *Computer Standards & Interfaces, 82*, 103610.
23. Ijaz, Q., Asghar, H., & Ahsan, A. (2016). *Exploratory study to investigate the correlation and contrast between ISO 9001 and CMMI framework: Context of software quality management.* Paper presented at the 2016 Sixth International Conference on Innovative Computing Technology (INTECH).
24. Jeffries, R. (2001). What is extreme programming. *XP magazine, 11*.
25. Kurapati, N., Manyam, V. S. C., & Petersen, K. (2012). *Agile software development practice adoption survey.* Paper presented at the International Conference on Agile Software Development.
26. Lee, I., Sokolsky, O., Chen, S., Hatcliff, J., Jee, E., Kim, B., . . . Roederer, A. (2011). Challenges and research directions in medical cyber–physical systems. *Proceedings of the IEEE, 100*(1), 75-90.
27. Lee, M.-C., & Chang, T. (2006). Applying TQM, CMM and ISO 9001 in knowledge management for software development process improvement. *International Journal of Services and Standards, 2*(1), 101-115.
28. Leveson, N. G., & Weiss, K. A. (2009). Software system safety. In *Safety Design for Space Systems* (pp. 475-505): Elsevier.
29. Machado, E. M. M., Mexas, M. P., Meza, L. A., & de Oliveira, I. A. M. (2022). IMPLEMENTATION PROCESSES OF QUALITY MULTI MODELS: ISO 9001, CMMI-DEV ML3, AND MPT. BR ML3 IN SMALL AND MEDIUM-SIZED ENTERPRISES (SMES) OF INFORMATION TECHNOLOGY. *Economic and Social Development: Book of Proceedings*, 29-38.
30. Mahanti, A. (2006). Challenges in enterprise adoption of agile methods-A survey. *Journal of Computing and Information technology, 14*(3), 197-206.
31. Montoni, M. A., Rocha, A. R., & Weber, K. C. (2009). MPS. BR: a successful program for software process improvement in Brazil. *Software Process: Improvement and Practice, 14*(5), 289-300.
32. Mutafelija, B., & Stromberg, H. (2003). *Systematic process improvement using ISO 9001: 2000 and CMMI*: Artech House.
33. Mylopoulos, J. (1992). Conceptual modelling and Telos. *Conceptual modelling, databases, and CASE: An integrated view of information system development*, 49-68.
34. OMG. (2011). Business Process Model and Notation (BPMN) Specification. *Version 2.0*.
35. Organization, I. S. (2000). ISO 9001:2000.

36. Organization, I. S. (2015). ISO 9001:2015.
37. Ozkaya, I. (2021). Can We Really Achieve Software Quality? *IEEE software, 38*(3), 3-6.
38. Pardo-Calvache, C. J., Chilito-Gómez, P. R., Viveros-Meneses, D. E., & Pino, F. J. (2019). Scrum+: A scaled Scrum for the agile global software development project management with multiple models. *Revista Facultad de Ingeniería Universidad de Antioquia*(93), 105-116.
39. Pastor, O., España, S., Panach, J. I., & Aquino, N. (2008). Model-driven development. *Informatik-Spektrum, 31*(5), 394-407.
40. Pino, F. J., García, F., & Piattini, M. (2008). Software process improvement in small and medium software enterprises: a systematic review. *Software Quality Journal, 16*(2), 237-261.
41. Poth, A., Sasabe, S., Mas, A., & Mesquida, A. L. (2019). Lean and agile software process improvement in traditional and agile environments. *Journal of Software: Evolution and Process, 31*(1), e1986.
42. Qumer, A., & Henderson-Sellers, B. (2008). A framework to support the evaluation, adoption and improvement of agile methods in practice. *Journal of Systems and Software, 81*(11), 1899-1919.
43. Rodríguez, M., & Piattini, M. (2012). *Systematic review of software product certification.* Paper presented at the 7th Iberian Conference on Information Systems and Technologies (CISTI 2012).
44. Runeson, P., Host, M., Rainer, A., & Regnell, B. (2012). *Case study research in software engineering: Guidelines and examples*: John Wiley & Sons.
45. Ruy, F. B., de Almeida Falbo, R., Barcellos, M. P., & Guizzardi, G. (2014). *An Ontological Analysis of the ISO/IEC 24744 Metamodel.* Paper presented at the FOIS.
46. Schwaber, K., & Beedle, M. (2002). *Agile software development with Scrum* (Vol. 1): Prentice Hall Upper Saddle River.
47. SEI. (2010). *CMMI for Development v1. 3, CMU/SEI-2010-TR-033*.
48. Silva, F. S., Soares, F. S. F., Peres, A. L., de Azevedo, I. M., Vasconcelos, A. P. L., Kamei, F. K., & de Lemos Meira, S. R. (2015). Using CMMI together with agile software development: A systematic review. *Information and Software Technology, 58*, 20-43.
49. Singh, A., Singh, K., & Sharma, N. (2012). Managing knowledge in agile software development. *International Journal of Advanced Computer Science and Applications (IJACSA), 2*(4).
50. Team, C. P. (2006). CMMI for Development, version 1.2.
51. Unterkalmsteiner, M., Gorschek, T., Islam, A. M., Cheng, C. K., Permadi, R. B., & Feldt, R. (2011). Evaluation and measurement of software process improvement—a systematic literature review. *IEEE Transactions on Software Engineering, 38*(2), 398-424.
52. Valdés, G., Visconti, M., & Astudillo, H. (2011). *The tutelkan reference process: A reusable process model for enabling SPI in small settings.* Paper presented at the European Conference on Software Process Improvement.
53. Walz, J., & Caroll, S. (2011). *IEEE Std 730 Software Quality Assurance: Supporting CMMI-DEV v1. 3, Product and Process Quality Assurance*. Retrieved from
54. Weske, M. (2007). *Business process management architectures*: Springer.

55. Wohlin, C., Runeson, P., Höst, M., Ohlsson, M. C., Regnell, B., & Wesslén, A. (2012). *Experimentation in software engineering*. Springer Science & Business Media.

Figures

Figure 1

General Schema for a quality assessment and certification process

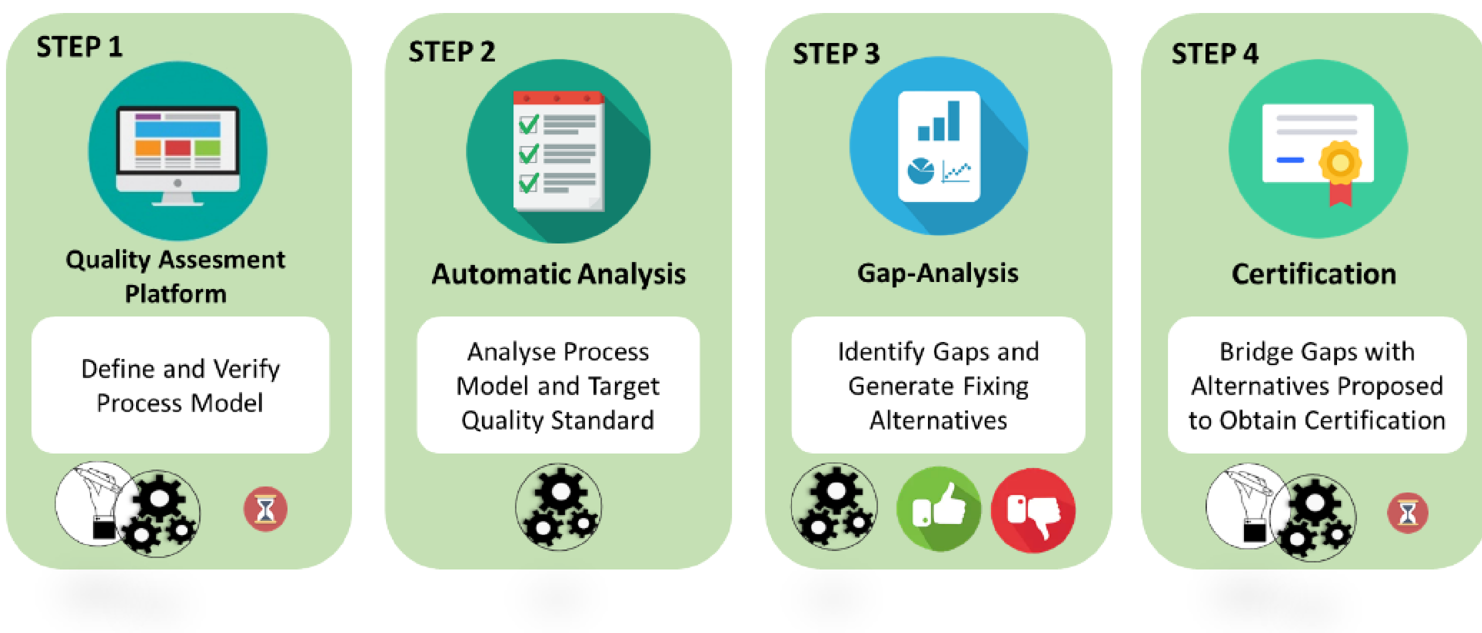

Figure 2

Quality assessment and certification process with the contributions proposed

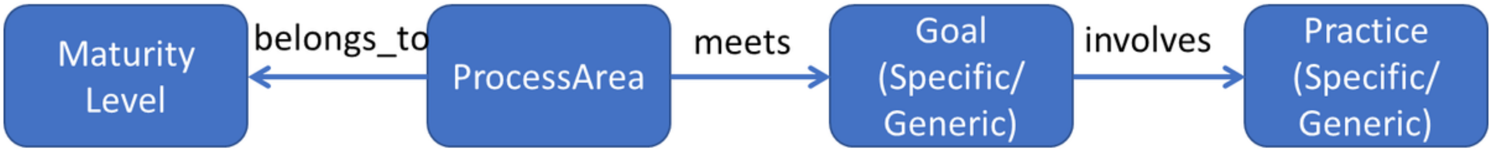

Figure 3

Simplified relationships between CMMi concepts – Staged Use

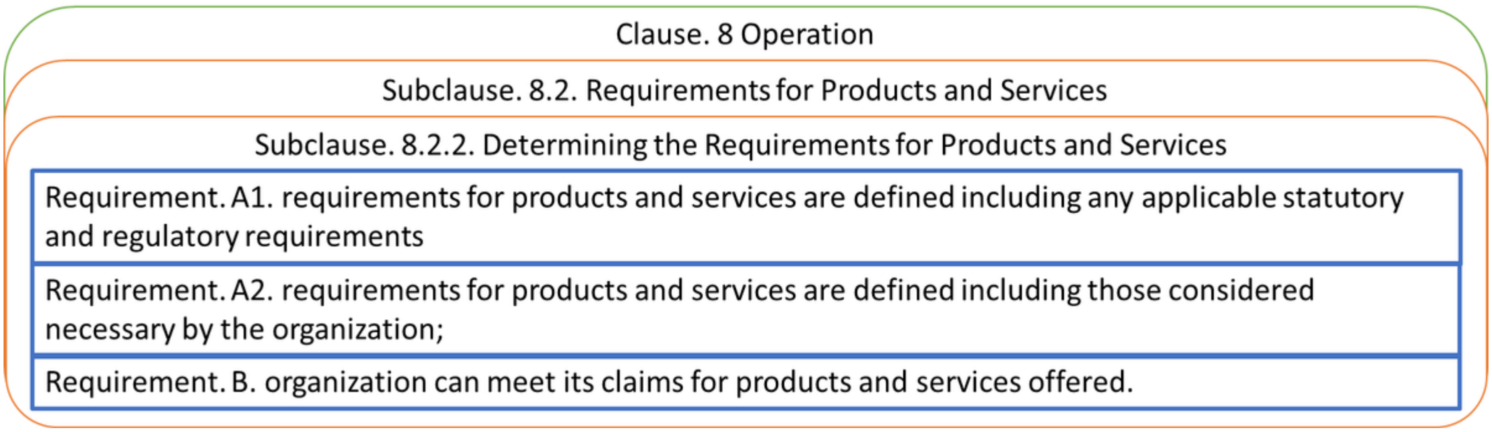

Figure 4

Schema of ISO9001 concepts – requirements for subclause 8.2.2

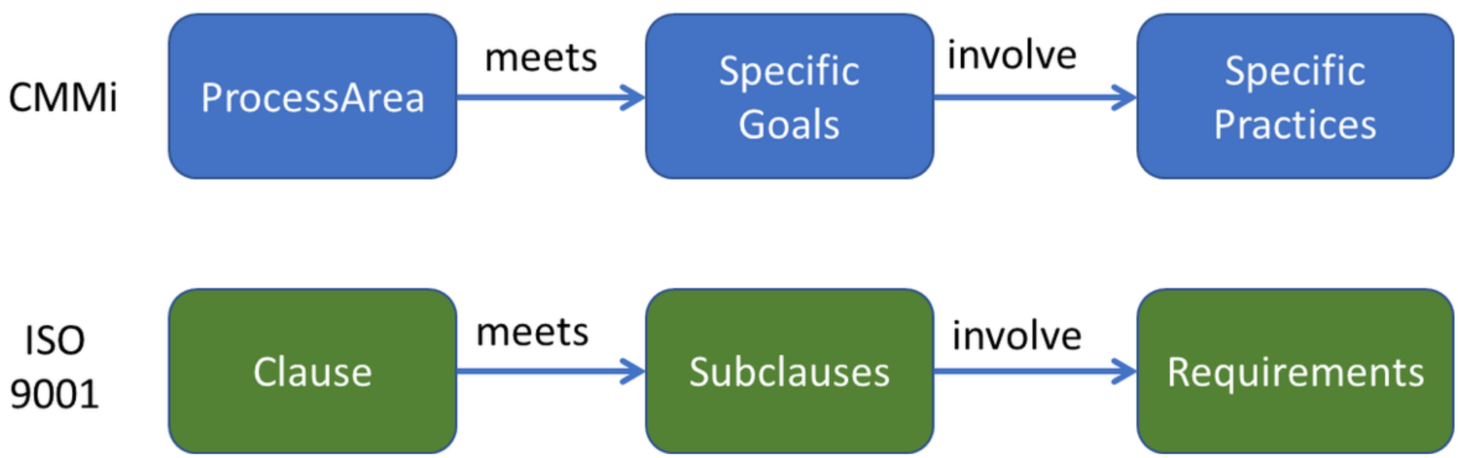

Figure 5

CMMi and ISO 9001 Structural Equivalences

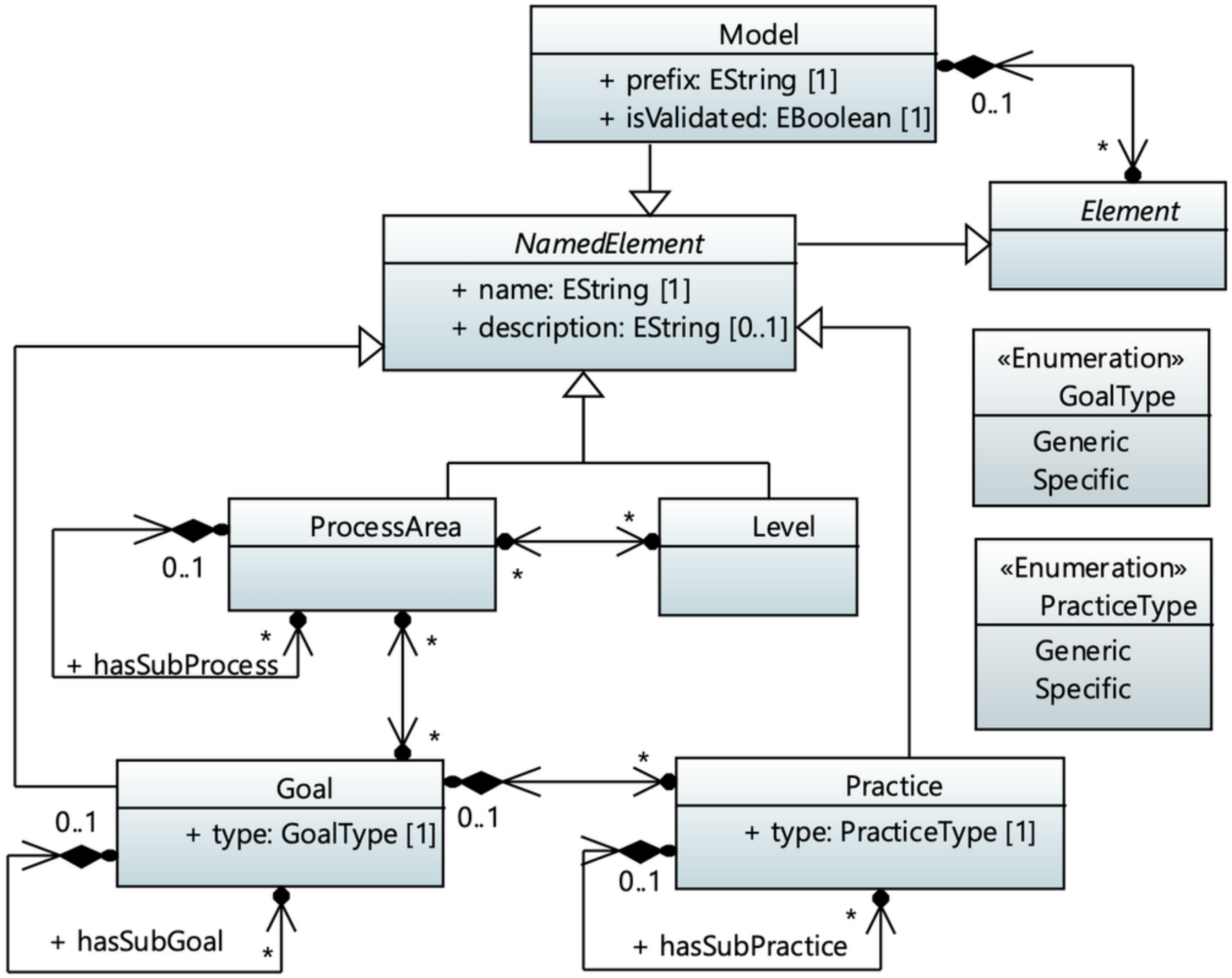

Figure 6

Metamodel of Quality Standards

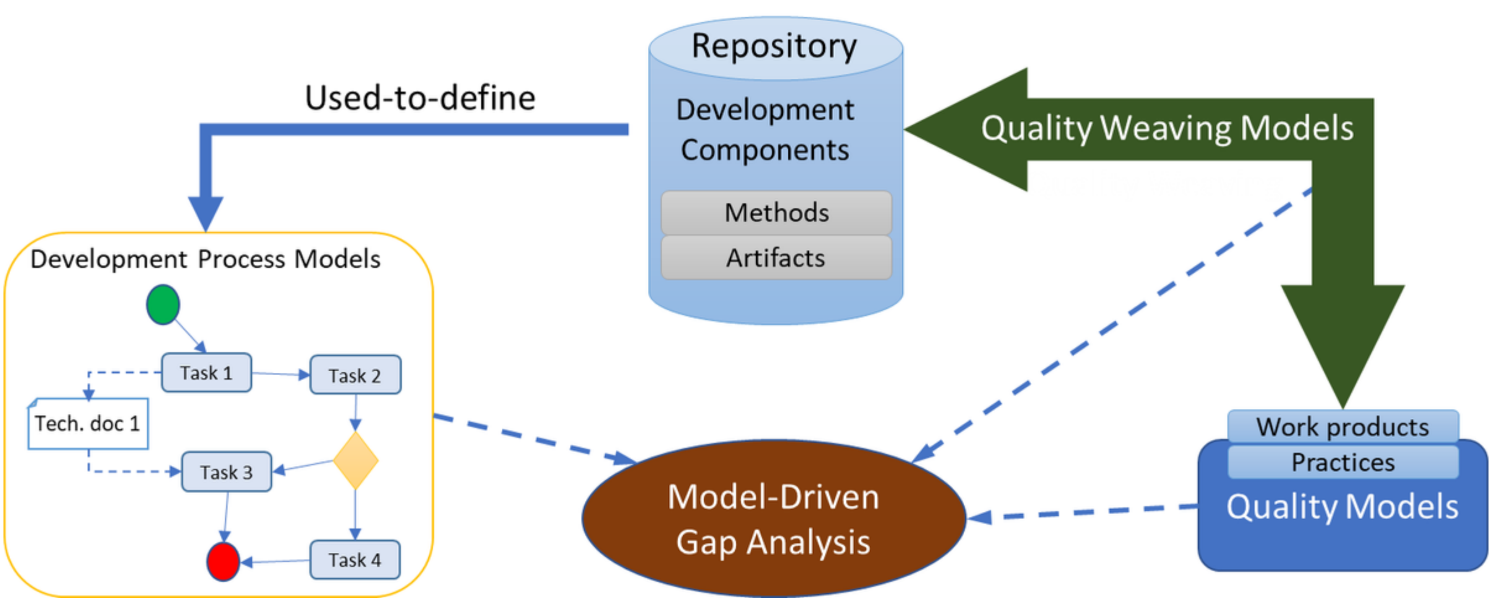

Figure 7

Model-Driven Gap Analysis General Conceptual Schema

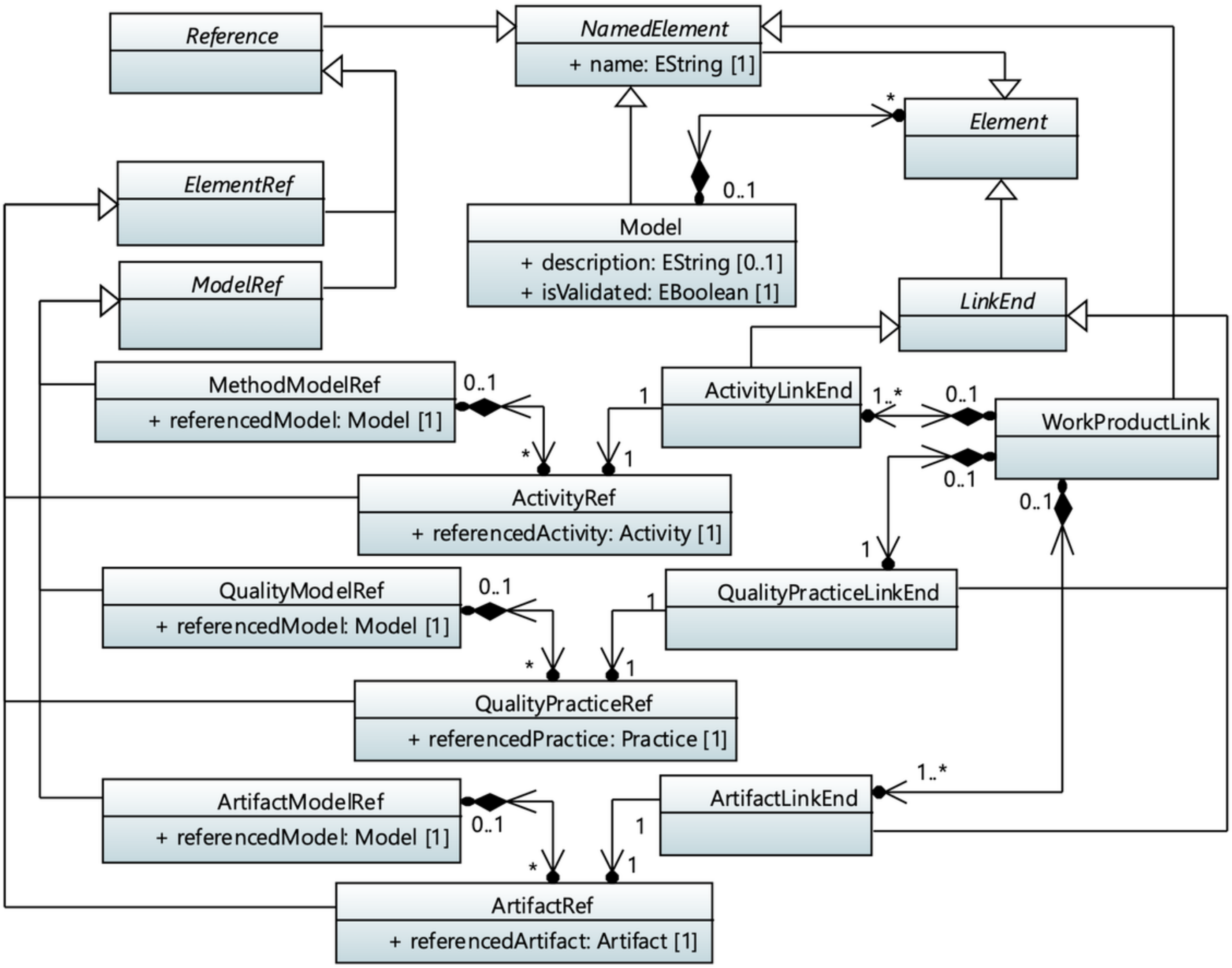

Figure 8

Quality Weaving Metamodel

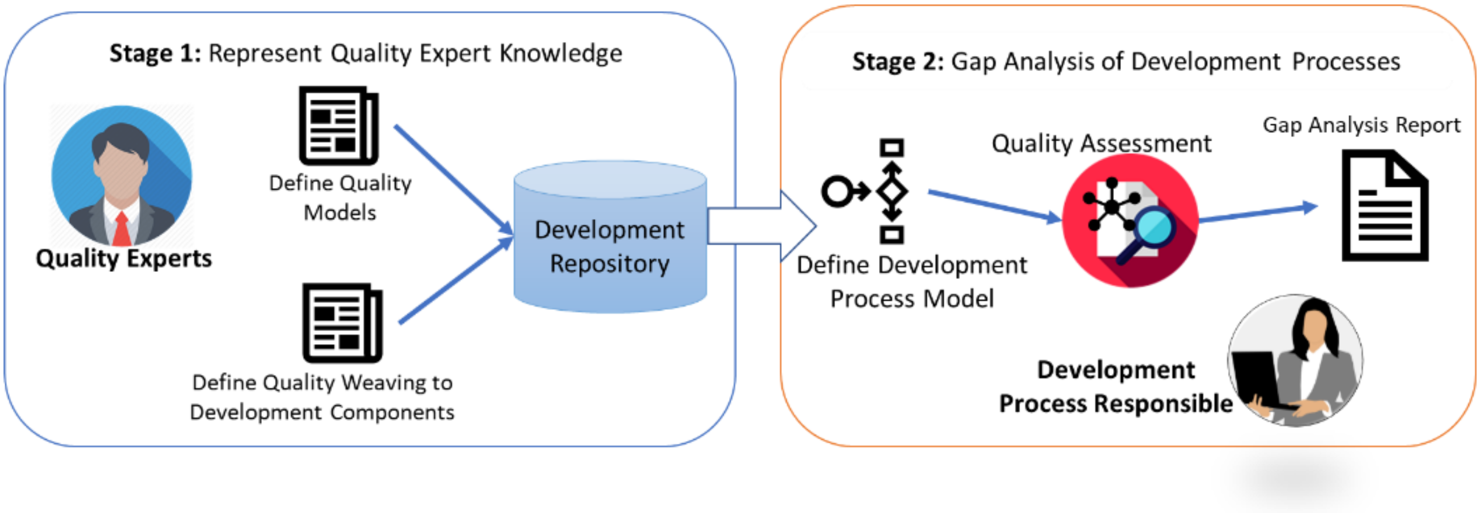

Figure 9

Workflow for Model-Driven Gap Analysis

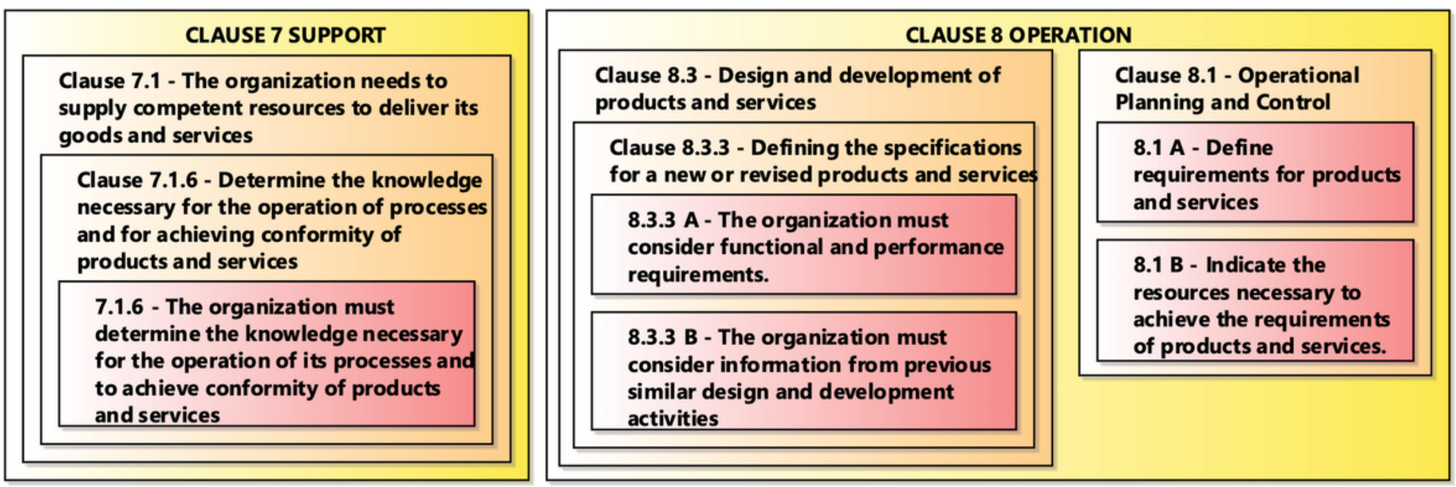

Figure 10

Example quality model defined for the ISO 9001 standard

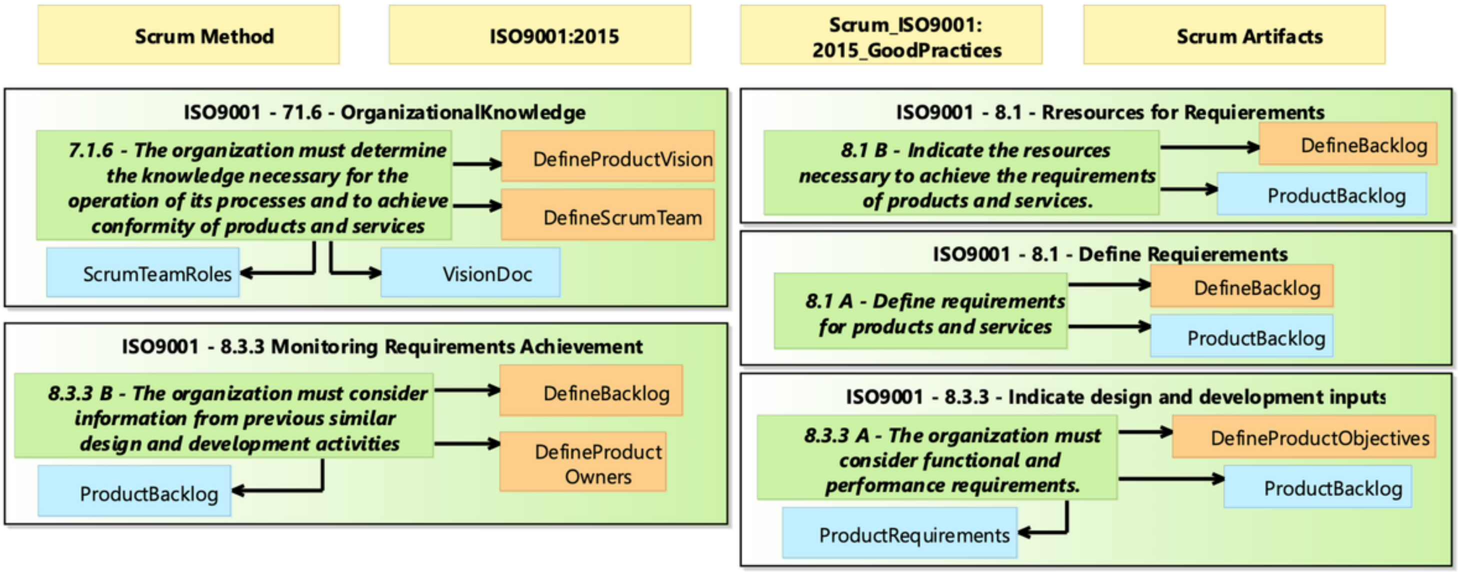

Figure 11

Quality weaving model for ISO9001 with the Scrum Agile Method

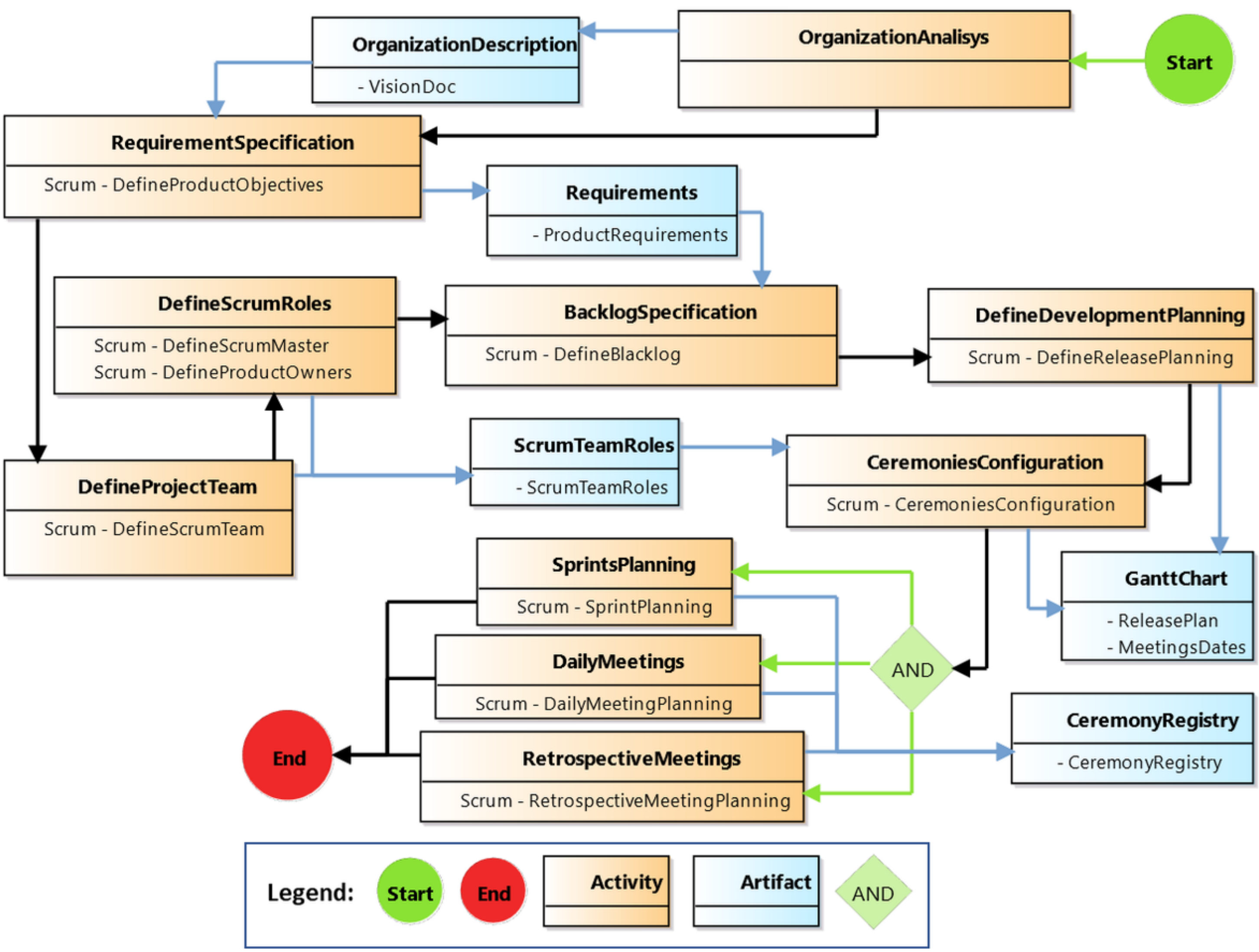

Figure 12

Example agile process defined as input for the quality assessment

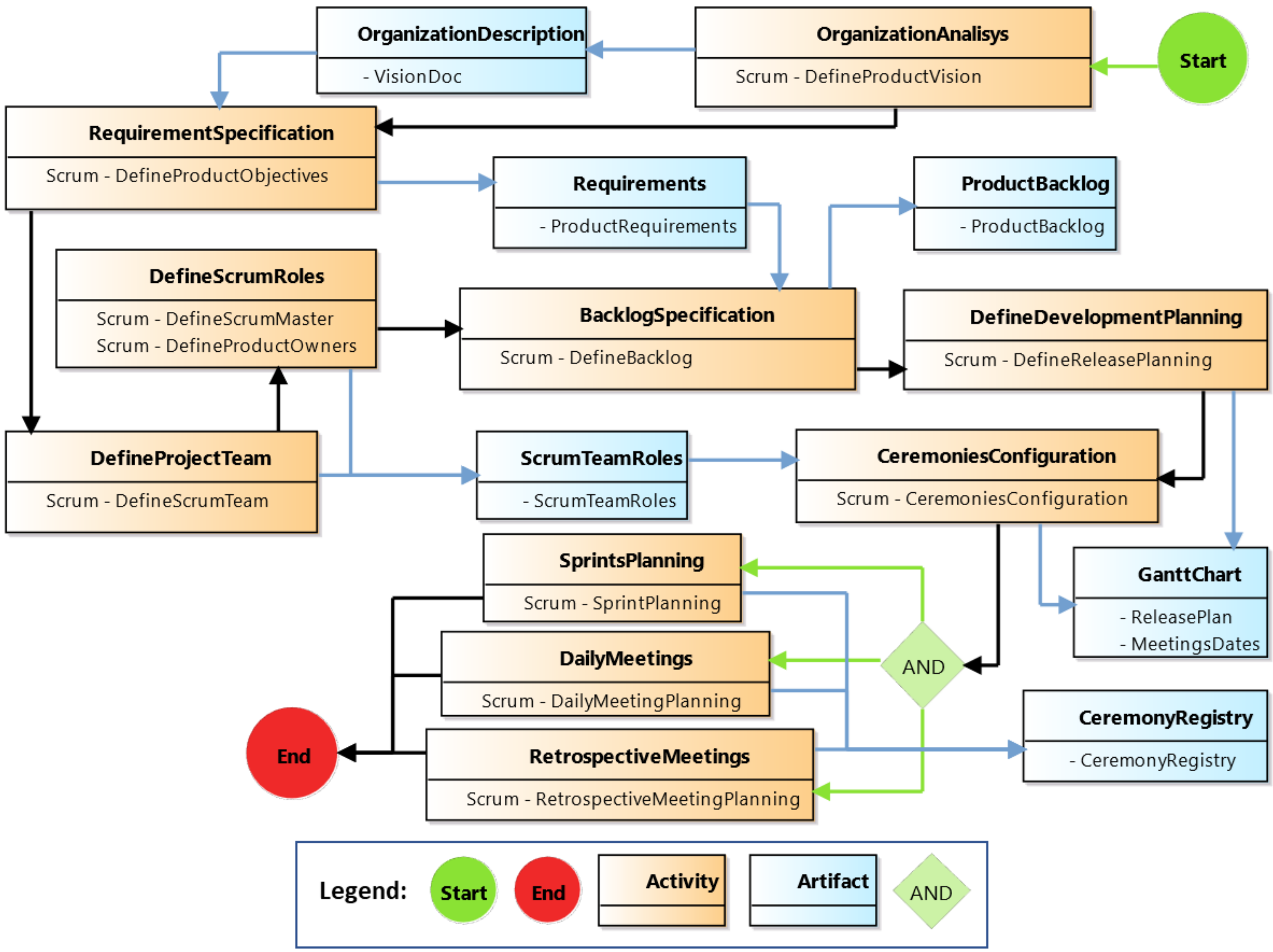

Figure 13

Development process improved with the ISO 9001 Gap Analysis