



D3.1 Dashboard Specification v1.0

Programme	H2020		
Funding scheme	RIA - Research and Innovation action		
Topics	ICT-10-2016 - Software Technologies		
Project number	732253		
Project name	Quality-aware rapid software development		
Project duration	1 st November 2016 – 31 st October 2019		
Project website	www.q-rapids.eu		
Project WP	WP3 – Strategic Decision Making Dashboard		
Project Task	Task 3.1 – <i>Dashboard Specification</i>		
Deliverable type	X	R	Document, report
		DEM	Demonstrator, pilot, prototype
		DEC	Websites, patent filings, videos, etc.
		OTHER	Material that does not belong to any specified category
		ETHICS	Ethics requirement
Contractual delivery	30/04/2017		
Delivered	30/04/2017		
Responsible beneficiary	UPC		
Dissemination level	X	PU	Public
		CO	Confidential, only for members of the consortium (including the Commission Services)
		EU-RES	Classified Information: RESTREINT UE (Commission Decision 2005/444/EC)
		EU-CON	Classified Information: CONFIDENTIEL UE (Commission Decision 2005/444/EC)
		EU-SEC	Classified Information: SECRET UE (Commission Decision 2005/444/EC)



Scheduled versions		
Version no.	Date	Month
1.0	30/04/2017	M6
2.0	31/10/2017	M12
3.0	31/10/2018	M24
4.0	31/10/2019	M36



Version history			
Version no.	Date	Description	Author
v0.0	22/12/2016	Outline	L. López (UPC)
v0.1	28/03/2017	Outline reviewed	All authors (UPC)
v0.2	28/03/2017	Initial content for some sections and/or annexes (Overall description, Dashboard requirements, Annex requirements documentation)	L. López (UPC)
v0.3	02/04/2017	Initial content for ontology section	C. Gómez (UPC)
v0.4	03/04/2017	Initial content for ontology, state of the practice (use cases), conceptual model	C. Gómez, C. Ayala, M. Oriol (UPC)
v0.5	07/04/2017	State-of-the practice & requirements added	X. Burgués, L. López (UPC)
v0.6	11/04/2017	Complete draft – except definition of terms	L. López (UPC)
v0.6.5	12/04/2017	General revision	X. Franch (UPC)
v0.7	13/04/2017	Version sent for internal revision	L. López (UPC)
v0.8	24/04/2017	State of the art included. State of the practice (tools) & Ontology reviewed	C. Gómez, X. Burgués, L. López (UPC)
v0.9	28/04/2017	Requirements reviewed	L. López (UPC)
v0.95	28/04/2017	General review	L. López (UPC)
v1.0	30/04/2017	Finalisation and upload	X. Franch (UPC)
V1.1	01/10/2017	Scheduled version table reviewed (wrong dates for v2.0 and v3.0)	L. López (UPC)
V1.2	15/10/2017	No changes in the Dashboard Specification explicitly stated in the executive summary	L. López (UPC)
V1.3	30/10/2017	Typo in the executive summary	L. López (UPC)
V2.0	30/10/2017	Finalisation and upload	X. Franch (UPC)



Authors	
Organization	Name
UPC	Lidia López
UPC	Cristina Gómez
UPC	Claudia Ayala
UPC	Xavier Burgués
UPC	Marc Oriol
UPC	Xavier Franch
Reviewers	
UOULU	Pilar Rodríguez
BITTIUM	Jari Partanen

Disclaimer

The work associated with this report has been carried out in accordance with the highest technical standards and the Q-Rapids partners have endeavoured to achieve the degree of accuracy and reliability appropriate to the work in question. However, since the partners have not control over the use to which the information contained within the report is to be put by any other party, any other such party shall be deemed to have satisfied itself as to the suitability and reliability of the information in relation to any use, purpose or application.

Under no circumstances will any of the partners, their servants, employees or agents accept any liability whatsoever arising out of any error or inaccuracy contained in this report (or any further consolidation, summary, publication or dissemination of the information contained within this report) and/or the connected work and disclaim all liability for any loss, damage, expenses, claims or infringement of third party rights.



Definition of the key terms	
Functional requirement	It defines a functionality that the system to be implemented shall offer.
Non-functional requirement	In the Q-Rapids project, it is used as synonymous of Quality Requirement.
Non-technical requirement	Requirements that does not refer directly to the intrinsic quality of software, but to the context of the system under analysis. Usually, it addresses economic, political or managerial issues.
Quality requirement	It is a requirement that states conditions on, and analyse compliance of, software quality.
Theme	In agile software development, it is a collection of user stories.
User story	In agile software development, it is a simple narrative illustrating the user goals that a software function will satisfy.

Abbreviations	
BI	Business Intelligence
BSC	Balance Scorecard
DoA	Description of Action
KPI	Key Performance Indicator
PO	Project Officer
QR	Quality Requirement
SotP	State of the Practice



Contents

The list of tables	8
The list of figures	9
Executive summary.....	10
1 Introduction.....	11
1.1 Motivation	11
1.2 Intended audience.....	11
1.3 Scope	11
1.4 Relation to other deliverables	11
1.5 Structure of the deliverable	11
2 Strategic Dashboard Description.....	12
2.1 Product Perspective.....	12
2.2 Product Features	12
2.3 User Documentation	12
3 Dashboard Requirements.....	13
3.1 List of Roles.....	14
3.2 Requirement Sources	14
3.2.1 State of the Art	14
3.2.2 State of the Practice: Use Cases	15
3.2.3 State of the Practice: Tools.....	16
3.3 Strategic Dashboard Requirements: Summary	17
3.4 Constraints.....	19
4 Ontology	20
4.1 Ontology Concepts	20
4.2 Ontology Validation.....	22
5 Dashboard Conceptual Model.....	23
5.1 Structural Schema	23
5.2 Scenarios of Use	23
6 User Interface Design	24
Conclusion	26
References.....	27
Annex A. Semi-structured interview conducted at Q-Rapids industrial partners’ premises.....	29



Annex B. Strategic Dashboard Requirements.....	31
Annex C. Ontology	36
Annex E.1. Concepts of Quality Assessment Package	36
Annex E.2. Concepts of Quality Aware Rapid Software Development Process Package	37
Annex E.3 Concepts of Strategic Decision Making Package	39
Annex D. Mock-ups	44



The list of tables

Table 1 Features requested by the use cases.....	16
Table 2: BI and Software Code Quality Tools features	17
Table 3 Strategic Dashboards Requirements	18
Table 4 Strategic Dashboard user stories (theme Analysis)	31
Table 5 Strategic Dashboard user stories (theme Characterisation)	31
Table 6 Strategic Dashboard user stories (theme Compatibility)	32
Table 7 Strategic Dashboard user stories (theme Interaction)	32
Table 8 Strategic Dashboard user stories (theme Management)	32
Table 9 Strategic Dashboard user stories (theme Reporting)	32
Table 10 Strategic Dashboard user stories (theme Security)	33
Table 11 Strategic Dashboard user stories (theme Usability)	33
Table 12 Strategic Dashboard user stories (theme Visualisation).....	33
Table 13. Strategic Dashboard non-technical requirements.....	35



The list of figures

Figure 1: The Q-Rapids Framework (source: Q-Rapids DoA).....	12
Figure 2: Dashboard Requirements gathering process	13
Figure 3: Strategic Dashboard requirements per type	18
Figure 4: Strategic Dashboard functional requirements	19
Figure 5: Strategic Dashboard quality requirements	19
Figure 6: UML Package Diagram for the Strategic Dashboard Ontology	21
Figure 7: Concepts related to the Strategic Indicator concept	23
Figure 8: Concepts related to the Decision concept	24
Figure 9: Screenshot of the Strategic Dashboard layout mock-up.....	25
Figure 10: Screenshot of “View Strategic Indicators” in grid view.....	44
Figure 11: Screenshot of “View Strategic Indicators” in tabular view	45
Figure 12: Screenshot of “New Event” notification	45
Figure 13: Screenshot of “New quality requirements”	46
Figure 14: Screenshot of Taking actions on Quality Requirements	47
Figure 15: Screenshot of Apply actions	48



Executive summary

This deliverable contains the Strategic Dashboard tool specification. The document includes a description of the software system, including the requirements (functional requirements and quality requirements), a set of scenarios of use that help to understand the behaviour of the system and an initial version of the dashboard layout using story boards and mock-ups.

The first version of this document (v1.0) contained the dashboard specification including the requirements elicited from the project partners.

The second version of this document (v2.0) does not include any change in the dashboard specification described in the first version. The proof-of-concept of the Q-Rapids platform, which will be provided at M15, will contain a subset of user stories related to the theme *Visualisation* (Table 12). Concretely the ones referring to the quality model visualisation (Strategic Indicators, Factors and Metrics).



1 Introduction

1.1 Motivation

Before starting the development of any software product, its characteristics need to be specified. This applies to one of the core components of the Q-Rapids solution: the Strategic Dashboard. The motivation of this document is, thus, to specify the Q-Rapids Strategic Dashboard as starting point for its development.

1.2 Intended audience

The natural audience of this document are the Strategic Dashboard developers, who need to use this document as a basis for their development. Furthermore, any other project team member can use the document to be informed about the characteristics of this tool. As the dashboard development will be iterative, by having access to the current status of the expected functionalities (functional requirements) and behaviour (quality requirements), they can contribute to the next iterations of the development process. This document will be send to the PO and the reviewers assigned to the Q-Rapids. Finally, as this specification is a public document, it will be accessible by any person interested on the Strategic Dashboard.

1.3 Scope

The scope of this document is the full Q-Rapids project, in the Work Package 3 along its entire timeline. Three updates of this document are planned for months M12, M24 and M36, including the specification used for each of the three main development milestones (M15, M24 and M33).

1.4 Relation to other deliverables

This is a document related to the development of the Strategic Dashboard. Some of the information managed in the dashboard will come from the data gathered by Work Package 1, thus the deliverable is somehow related to *D1.1 - Data gathering and analysis specification*. Not at this initial stage, but when the dashboard requirements will be refined, we will have some data analysis requirements for Work Package 1, that will be reflected in future versions of D1.1.

As part of the development activities in the Q-Rapids project, the Strategic Dashboard development is regulated by Work Package 4. We are using the tools and the protocols described in *D4.1 - Development environment and methodology*. The dashboard requirements will be managed in the Q-Rapids backlog, described in *D4.2 – Product Backlog*.

1.5 Structure of the deliverable

This deliverable is organized into the following sections. Section 2 includes the overall description of the Strategic Dashboard system. Sections 3, 4 and 5 conform the dashboard specification. Section 3 includes the details about the requirements, Section 4 describes the ontology used to describe the concepts that will be used by the system, and Section 5 describes the conceptual model. Section 6 hints the look and feel of the component using story boards and mock-ups.

The information included in the different sections of this document is complemented with some annexes. In order to elicit requirements, we visited the partners that provide the use cases of the project as described in Annex A. The complete list of requirements, documented as user stories, is included in Annex B. Annex C includes the detailed set of concepts defined in the ontology to be used by the Strategic Dashboard. Finally, Annex D includes the details of the story boards and mock-ups.



2 Strategic Dashboard Description

2.1 Product Perspective

The Strategic Dashboard is a component meant to provide support to decision-makers, aggregating the information automatically gathered from several sources into strategic indicators. Decision-makers can then use this dashboard to feed the rapid development process.

Figure 1 presents the complete Q-Rapids framework. The Strategic Dashboard is represented at the bottom of the figure (*Quality-aware decision making dashboard*). As depicted in the figure, the dashboard is fed with data provided by the *Data mining engine*, and is managing the constraints that the decision-maker needs to take into account when s/he makes decisions related to quality requirements (QRs) and their consideration in the product backlog.

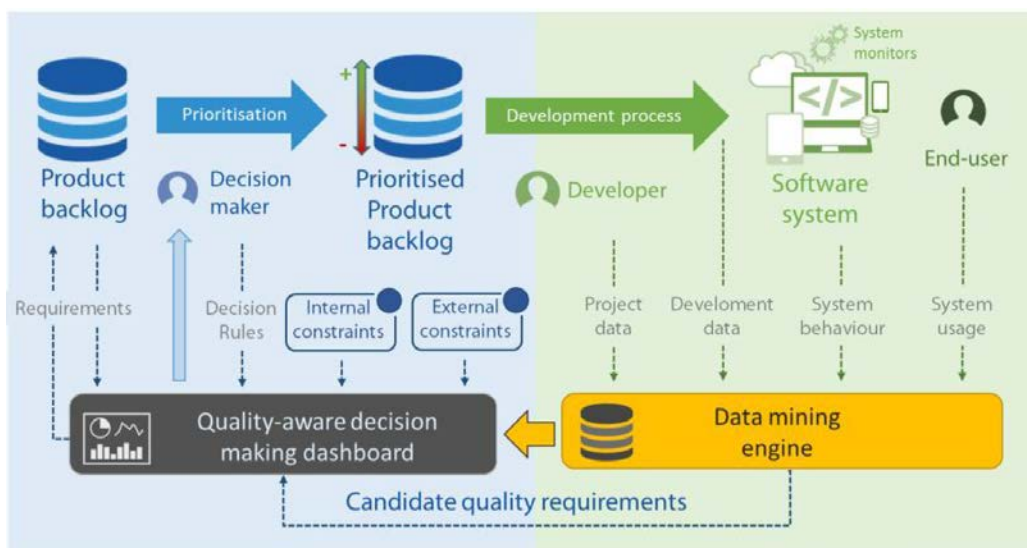


Figure 1: The Q-Rapids Framework (source: Q-Rapids DoA)

2.2 Product Features

The main functionality of the Strategic Dashboards is to present decision-makers aggregated strategic indicators in a concise, informative and friendly way. In the Q-Rapids project, according to its objectives, the strategic indicators will be related to QRs and their management in rapid software development processes.

Besides this natural data visualization functionality, the dashboard will also include some advanced techniques to give as much support as possible to decision-makers. These functionalities are summarised as:

- To anticipate the violation of strategic indicators based on the evolution of their value along time.
- To offer elaborated techniques for exploring alternatives to be followed in the software process based on their impact on the key indicators.
- To propose mitigation actions and contingency plans when violations or other type of deviations are predicted or detected.

2.3 User Documentation

As part of the dashboard deployment, we are going to deliver a user guide and an integration guide including all the details to integrate the dashboard result in other tools in the context of the project use cases.



3 Dashboard Requirements

Figure 2 presents the Strategic Dashboard requirement gathering process. The process includes two steps: requirements elicitation and requirements refinement. We started with an initial requirements elicitation task from several sources, followed by several iterations for refining them into system features. The initial set of requirements were identified directly from: the Q-Rapids DoA, a state of the art and a state of the practice. For the state of the practice, we conducted semi-structured interviews with representatives of the four Q-Rapids industrial partners. The results from the interviews were complemented with the analysis of some business intelligence (BI) and software quality measurement commercial tools. Section 3.2 includes the details related to each source. The requirements refinement phase is defined as an iterative process following the lean cycle (build, measure and learn). This refinement process will be conducted iteratively along the project lifecycle. For each iteration, a set of user stories are selected from the backlog and they are discussed and the features to be developed are included in the backlog for the next development iteration. During this stage, the refinement of the Strategic Dashboard user stories eventually will produce some user stories to be included in the data gathering backlog (WP1). Specifically, the user stories that require some data analysis that will be performed by the data mining engine. Annex D include the first built-measured-learn iteration for the first two user stories (*View Strategic Indicators* and *Assess a new Quality Requirement*).

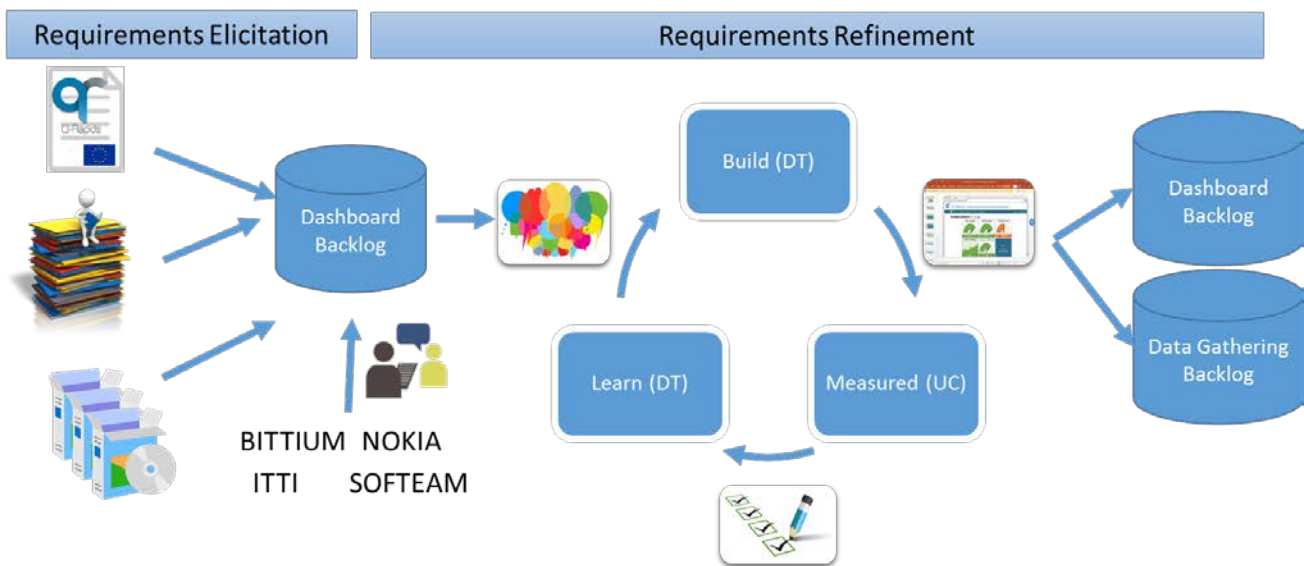


Figure 2: Dashboard Requirements gathering process

We have classified the user stories related to the Strategic Dashboard functionality in the following themes:

- **Analysis:** analysis of the data that comes with a result useful to the user (e.g., new QR candidate, violation prediction).
- **Characterisation:** management of the data used by the Strategic Dashboard (e.g. product, measures).
- **Interaction:** functionality that needs some feedback from the user
- **Management:** Strategic Dashboard administration functionality (e.g. user management).
- **Reporting:** report generation from the Strategic Dashboard.
- **Visualisation:** display of information.



For the classification of the quality requirements, we have used the characteristics included in the ISO/IEC 25010 quality model [ISO/IEC 25010:2011]: Functional Suitability, Performance Efficiency, Compatibility, Usability, Reliability, Security, Maintainability and Portability.

Besides the functional and the quality requirements, the Q-Rapids project milestones includes the following non-technical requirements [Palomares et al. 2012] related to the Strategic Dashboard releases and documentation. These requirements has not been included in the backlog as user stories.

- The proof-of-concept version should be released by the end of February 2018 (M15).
- The consolidated version should be released by the end of November 2018 (M24).
- The marketable version should be released by the end of August 2019 (M33).
- A user guide will be produced, describing how to use the system.
- A technical documentation will be produced, including all the necessary information to integrate the system in other software products.

3.1 List of Roles

The Strategic Dashboard natural users are people making decisions related to product requirements or features. The roles making these kind of decisions in rapid/agile software development processes are usually Product Managers and Product Owners. Additionally, we can include as target users other higher level such roles such as R&D release managers (when they are responsible of including features in the backlog).

As we are also applying an agile software development methodology, we do not discard that new roles can be considered when we develop the different features in the different development iterations. Because of the interest of some of the project use cases in giving visibility and transparency to the development and decision making processes, it is possible that the dashboard will be open to the development team. Some partner also mentioned that the functionalities included can be interesting for Enterprise Architects.

3.2 Requirement Sources

3.2.1 State of the Art

The goal of the Q-Rapids Strategic Dashboard is to support strategic decision making processes by providing strategic indicators in the context of quality requirements in agile and rapid software development projects. We reviewed the literature to find out what is the current state of the art in this field. Basically, we are interested in identifying the concepts of strategic decision making, relevant for decision-makers, and mapping those concepts to the context of software quality requirements in agile and rapid software development. We used the knowledge acquired reviewing the literature as one of the main inputs for the definition of the ontology of the strategic decision making package provided in Section 4.

There is an extensive literature on how to model and measure the strategy of a company that may be applied to the context of software quality requirements in agile and rapid software development projects. We used the concepts defined in the literature to model and measure strategic indicators for supporting decision making in our context. The Balance Scorecard (BSC), proposed in [Kaplan et al. 1996], is a business framework used for describing and measuring an organization's strategy and for tracking the actions taken to improve the results. In this sense, the BSC proposes to define *strategic objectives* to achieve the vision of an organization, *key performance indicators* (KPIs) to measure those strategic objectives and *actions* that the organization should take to achieve the strategic objectives. BSC has been applied in the context of measuring



software quality¹. The Business Motivation Metamodel (BMM), proposed in [OMG-BMM], provides concepts for developing, communicating and managing business plans in organized manner. Specifically, the BMM defines concepts such as *strategy* and *goals*. In the same sense, the Business Intelligence Model² (BIM) [Barone et al. 2010] provides constructs for modelling business organizations at strategic level. In particular, concepts such as *actors*, *intentions*, *situations* and *KPIs* are defined.

KPI, as a way of monitoring, is a crucial concept that has received a lot of attention in the literature. There exists catalogues of KPIs for measure several aspects of the organization. For instance, the *scoro work management* software solution³ provides 16 essential project KPIs to track a project's performance (e.g. return of investment, overdue project tasks/crossed deadlines), *enfocussolutions*⁴ define KPIs for business analysis and project management (e.g. project stakeholder satisfaction index, number of milestones missed) and CBS⁵ (Center for Business Practices) enumerates a comprehensive list of measures of project management and value in the context of IT organizations (e.g. average time to repair a defect, alignment to strategic business goals). In [Maté et al. 2014] the authors present a systematic semi-automatic approach that performs a partial search guided by the KPIs of the company, generating queries required during the monitoring process to discover the existence of problems and where they are located.

Regarding the decision making process, the Decision Model and Notation (DMN) metamodel [OMG-DMN] provides the constructs that are needed to model decisions. DMN defines concepts such as *decision*, *decision-maker*, *decision-owner* and *knowledge requirement*.

3.2.2 State of the Practice: Use Cases

In order to investigate the state of the practice in the context of the use cases provided by the industrial partners of the project, we performed a set of visits to the premises of each one of them. The goal of such visits was to investigate the AS-IS situation and the requirements and expectations that the partners have on the Q-Rapids project. In particular, in this deliverable, we report the data gathered corresponding to the expected requirements over the dashboard that each partner stated.

We designed a semi-structured interview guide as an instrument to perform face-to-face interviews to gather data during our visits to the industrial partners' premises. In Annex A, we provide the whole interview guide used; the section of the interview guide that is related to this deliverable is Section 6 (Q-Rapids expectations).

We performed 12 interviews in total (each partner contributed with at least two or at most four respondents). Two or three researchers from the UPC team participated in each interview. Interviews lasted around 1 hour and were recorded and subsequently transcribed by a third party company. We also took relevant notes and requested for additional documentation to the partners during our visits. It helped us to complete our observations. These interviews have been a rich source of requirements: the 75% of the requirements reported in Annex B are gathered from the industrial partners.

We used content analysis to perform data analysis. Data analysis was performed by three researchers from the UPC team (two of them also participated in the interviews).

¹ <http://www.bscdesigner.com/bsc-for-software-quality-guide.htm>

² <http://www.cs.toronto.edu/~jm/bim/>

³ <https://www.scoro.com/blog/16-essential-project-kpis/>

⁴ <http://enfocussolutions.com/kpis-for-business-analysis-and-project-management/>

⁵ http://www.pmsolutions.com/audio/PM_Performance_and_Value_List_of_Measures.pdf



Table 1 Features requested by the use cases

Theme	Feature
Visualise	Current product quality and trend charts
Visualise	Different viewpoints depending on the role/user
Visualise	Increasing the visibility/transparency
Visualise	Evolution of the product quality through the development process
Usability	Easy to use (low learning curve) and easy to integrate in their current processes
Usability	Easy to configure

As a result of the use case analysis, and the consequent requirement gathering, we observed that they are particular concerned about how the data will be visualised and the system usability (see Table 1). We remark that increasing the visibility/transparency was not an explicit requirement for the system; however, if the system achieves to accomplish the other visualisation features, it will naturally follows, and this is why we are including it in the table. The use cases also showed interest in the features related to the *Interaction* theme, but only when we asked for them explicitly.

3.2.3 State of the Practice: Tools

At this first stage, our target was to gather a set of functionalities present in currently existing dashboards to stand as an input to the requirements elicitation process of our own dashboard. We chose two particular domains as the most relevant for the dashboard: Business Intelligence (BI) tools and software code quality tools. In particular, there is a lot of knowledge and experience in BI that, in essence, is very near to the activities we are aiming to support: collecting information from the production system, aggregate that information in several convenient ways and helping managers to understand, predict and decide.

In this first iteration of the deliverable, we decided to explore two tools of each domain. For software quality, we chose SonarQube⁶, Bitergia⁷ and Black Duck⁸ as candidates to analyse. For BI, we chose Tableau⁹ and Microsoft Power BI¹⁰, which have the highest degree of ability to execute and completeness of vision, thus located in the LEADERS area of Gartner's Magic Quadrant [Gartner 2017]. Ease of use for content consumers was the most-cited reason for customers choosing Microsoft Power BI. A strong community of partners, resellers and individual users extends Microsoft BI with prebuilt apps, visualizations and video tutorials. Microsoft Power BI is mainly being used for parameterized reports and dashboards. On the other hand, Tableau's core product strengths continue to be its intuitive interactive visualization and exploration and analytic dashboarding capabilities for almost any data source.

The next step was obtaining information about the dashboard functionalities of the selected tools. For each one, we obtained a subset of: demo version, documentation, information from the company and information from users. From this information we extracted the functionalities that may contribute to make a dashboard as the one we are engaged. They are shown in Table 2. We have classified the features using the themes defined for the dashboard. Most of the features correspond to the themes *Analysis*, *Characterisation*, *Compatibility*, *Interaction*, *Management* and *Visualisation*). There is one feature "Sharing/Following

⁶ <https://www.sonarqube.org>

⁷ <http://bitergia.com>

⁸ <https://www.blackducksoftware.com>

⁹ <https://www.tableau.com>

¹⁰ <https://powerbi.microsoft.com/en-us>



dashboards” that is not included as a requirement in the Strategic Dashboard; we will consider its inclusion in future iterations, maybe creating a new theme (*Sharing*).

Table 2: BI and Software Code Quality Tools features

Themes	Feature	Use in Q-Rapids
Visualise	Use of maps to display geographical data	Use of software architecture to attach data to the different product components, like a mind map, where the nodes would be the product components
Characterise	Managing dimensions and aggregation levels	Dimensions like component (levels package, module, class, ...), human resources (centre, team, person, ...), project, version, ...
Analysis	Roll up/Drill down	
Characterise	Managing measures	Measures like complexity, time to fix, ...
Visualise	Simultaneous display of functions to compare. Or tabular presentation of functions	Planned vs. actual, comparing projects, teams, ...
Visualise	Tabs to switch between dashboards	Tabs related to quality requirements
Interaction	What-If capabilities. For instance, changing measures recalculating values and functions	Adjusting user stories points, velocities, team human resources, ...
Interaction	Interactive dashboard elements / Navigable dashboards	Easy way to see consequences of changes and to reach desired view
Management	User management. Dashboard administration.	Developers, stakeholders, product owner, ...
Sharing	Sharing/Following dashboards.	Developers, stakeholders, product owner, ...
Analysis	Query creation support <ul style="list-style-type: none"> • Natural language • Navigating through measures and dimensions 	No particular comments on these functionalities, apart the interest of considering them as part of our dashboard.
Characterise	Wide catalogue of metrics	
Analysis	Filtering data	
Visualise	Wide catalogue of graphical representations	
Visualise	Mobile/Web/Desktop interface	
Visualise	Customizable dashboards	

3.3 Strategic Dashboard Requirements: Summary

The requirements have been documented as user stories and managed using the Redmine tool. We followed the protocol for the requirement documentation included in Deliverable D4.2.



From the different sources, we identified 40 functional requirements, 9 QRs and 5 non-technical requirements. The full list of requirements for the Strategic Dashboard is included in Annex B. Table 3 includes the themes we have defined to classify the Strategic Dashboard requirements, for each theme, the last two columns includes the information about the number of user stories related to the theme and the number that are gathered from the use cases. The numbers preceded by the character # are the IDs assigned for the user stories in the Redmine tool, for the WP3 subproject. As shown in Figure 3, most of the elicited requirements are functional requirements.

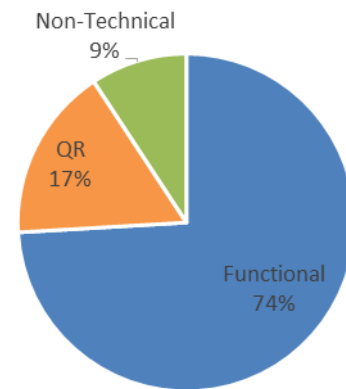


Figure 3: Strategic Dashboard requirements per type

Table 3 Strategic Dashboards Requirements

ID	#	Type	Subject	Number of req.	Number of Req. from UC
WP3-ANLS	#122	Functional	Analysis	6	3
WP3-CHAR	#123	Functional	Characterisation	5	3
WP3-COMP	#117	Quality	Compatibility	1	0
WP3-DOC		Non-technical	Documentation	2	0
WP3-INTR	#124	Functional	Interaction	4	4
WP3-MNG	#125	Functional	Management	2	0
WP3-REL		Non-technical	Release	3	0
WP3-REP	#116	Functional	Reporting	3	3
WP3-SEC	#118	Quality	Security	2	0
WP3-USA	#119	Quality	Usability	6	6
WP3-VIS	#121	Functional	Visualisation	20	20
Total number of requirements				54	39

Figure 4 shows that most of the functional requirements are related to the *Visualisation* theme, followed by *Analysis*, *Characterisation* and *Interaction*. Due to the system nature (a dashboard is visual by definition), the use cases are particularly interested in this aspect (20 from the 39 requirements elicited from the use cases). There are several user stories related to see trends charts (#67, #85, #90, #106) complementing the quality at specific point in time (#54, #103). The second cluster of user stories are related to the *Analysis* theme: referring to the monitoring quality (#58), suggesting new features and quality requirements (#60, #96), predicting violations (#97) and identifying mitigation activities (95). These features mainly correspond to the functionalities included in the DoA. The third related to *Characterisation* theme, the system should manage some information, such as constraints (#94) and milestones (#70). Like for the *Characterisation*, for the *Interaction*, the functionality is related to the what-if analysis included in the DoA (#52, #73, #87).

Many functional requirements come from, or are confirmed by, the results obtained in the state of the practice - tools analysis (Section 3.2.3). Especially in interaction (#52, #66), management (#126, #127) and visualisation themes (#56, #65). On the other hand, some functionalities found in the analysed tools have not been incorporated to the requirements at this stage, but deserve to be considered in future iterations. For instance, visualization of data attached to components (in a similar way to display data on maps) and sharing dashboards and navigating dashboards.



Most of the QRs elicited from the use cases are related to maximise the user acceptance of the new tool, concretely related to the *Usability* theme: the tool should be easy-to-use, easy-to-configure and easy-to-integrate with their processes (user stories #74, #78, #79, #81). As stated in the project, including the quality requirements from the very beginning of the development process increases the product quality and success. Therefore, we complemented the quality requirements with requirements related to security related to authentication (#99, #101).

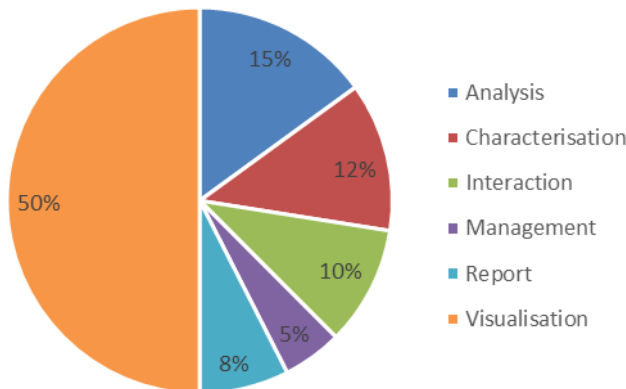


Figure 4: Strategic Dashboard functional requirements

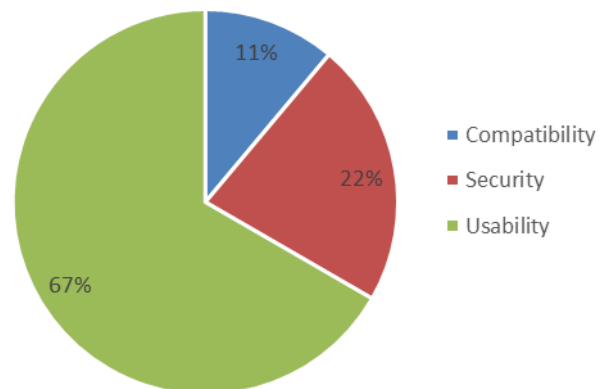


Figure 5: Strategic Dashboard quality requirements

According to the requirements documentation guidelines, the quality requirements can be documented as user stories or acceptance criteria depending on the scope of the requirement. For QRs referring to specific functionality, they need to be included as acceptance criteria of the corresponding user story. Currently, all the identified quality requirements apply to the whole system, therefore they all have been documented as separate user stories.

3.4 Constraints

At this stage, we have not identified any constraint.



4 Ontology

This section presents a first version of the ontology that defines the semantics of the domain concepts that will be managed by the Strategic Dashboard.

The ontology is being conceptualized following an iterative and incremental process based on Methontology [Fernández-López et al., 1997]. The activities performed to define the ontology were:

- *Definition of the ontology structure*: Identification of the ontology packages to group concepts with related semantics and to provide a namespace for the grouped elements.
- *Extraction of terms relevant for the Strategic Dashboard*: Terms including concepts, verbs, instances and properties were extracted from the DoA document and from the use cases analysis documentation. Other terms will be identified as the ontology construction process advance.
- *Concept definition*: For each relevant term and terms representing concepts identified in the previous step, we provide the following information:
 - **Description**: A brief explanation of the concept.
 - **Source(s) (only for concepts coming from an external source)**: The name of the source(s) and the reference(s) included in the References section.
 - **Example/s**: One or more examples that exemplify the concept in the context of the Q-Rapids project.
 - **Attributes (optional)**: Each attribute should be enumerated together with a short description. The multiplicity of the attribute is omitted if it is '1'. If it is coming from an external source, the reference to the source included in the References section will be included at the end of the attribute description.
 - **Associations (optional)**: The opposite ends of associations connected to the concept are also listed. If it is coming from an external source, the reference to the source included in the References section will be included at the end of the description.
 - **Generalizations (optional)**: Concepts that generalize the concept. If it is coming from an external source, the reference to the source included in the References section will be included at the end of the description.
 - **Constraints (optional)**: The well-formedness rules of the concept are defined as a (possibly empty) set of invariants, which must be satisfied by all instances of that concept for the model to be meaningful. If it is coming from an external source, the reference to the source included in the References section will be included at the end of the description.
 - **Synonyms (optional)**: Other names used for referring the concept.
 - **Remarks (optional)**: Comments or observations.

4.1 Ontology Concepts

The ontology is structured into three packages: Quality Assessment, Quality-Aware Rapid Software Development Process and Strategic Decision Making. Figure 6 shows a UML package diagram defining the three packages and the usage dependencies between them. Each usage dependency defines that a package requires another one for its full definition. For example, the Strategic Decision Making package requires concepts defined in the Quality Assessment and Quality Aware Rapid Software Development Process packages for its definition.

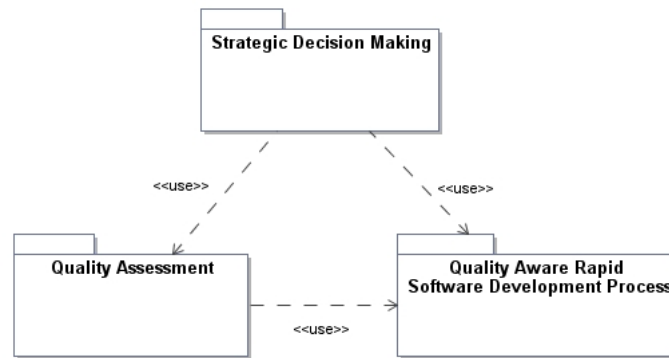


Figure 6: UML Package Diagram for the Strategic Dashboard Ontology

The Quality Assessment package includes the concepts related to the assessment of the level of software quality during development and runtime. The concepts defined in this package are all coming from the QUAMOCO quality meta-model [Wagner et al. 2012] (see Annex A.1 for their description):

- Aggregation
- Data Source
- Entity
- Factor
- Instrument
- Measure
- Process Factor
- Product Factor
- Quality Factor

The Quality-Aware Rapid Software Development Process package includes the concepts related to the development process focusing on the software life cycle integrating quality and functional requirements. The concepts defined in this package are based in some ISO standards ([ISO 9000:2005] [ISO 12207: 2008] [ISO 26515: 2012]), proposals for rapid [Ernst et al. 2012] [Fitzgerald-Bstol 2017][Greer-Ruhe 2004] and agile [Leffingwell, D. 2011], requirements engineering [Berander-Andrews 2005] and software acceptance [Wallace-Cherniavsky 1990] (see Annex A.2 for their description):

- Acceptance Criteria
- Agile Software Development
- Developer
- Epic
- Feature
- Feature Team
- Process
- Product Manager
- Product Owner
- Project Portfolio
- Quality
- Quality Requirement
- Rapid Software Development
- Release
- Requirement
- Requirements Prioritization
- Sprint
- Task
- Team backlog
- Tester
- User Story

The Strategic Decision Making package includes the concepts related to strategic quality-aware key indicators and to the process for supporting decision-makers to make strategic decisions. Some of the concepts defined in this package are based in the BIM language [Barone et al. 2010]] (see Annex A.3 for a description of these concepts):

- Factor
- Action
- Constraint
- Decision
- Decision-maker
- Decision Rule
- External Constraint
- Internal Constraint
- KPI
- KPI Evaluation
- Quality Requirement
- Quality Requirement Action
- Role
- Software Product



- Strategic Indicator

4.2 Ontology Validation

We plan to validate the ontology from the use cases. The goal of the validation is twofold: to confirm the concepts defined in the ontology and to refine or define new concepts. The validation will be performed following the subsequent steps:

- *Instantiating the ontology with the information provided by the use cases*: Identification of instances of or concrete evidence in support of the concepts of the ontology. For those concepts whose instances are not identified from the use cases, we will ask companies for additional information in order to confirm that the concepts are relevant.
- *Analysing the UC coverage*: The analysis of the information provided by the use cases may reveal concepts that are not currently defined in the ontology.



5 Dashboard Conceptual Model

5.1 Structural Schema

The structural schema defines the domain concepts of the Strategic Dashboard using the concepts defined in the ontology. Figure 7 and Figure 8 show two fragments of the structural schema using UML class diagrams. Concretely, Figure 7 shows the concepts related to the Strategic Indicator concept. A Strategic Indicator is an aspect that a company considers relevant for the decision making process during the software process development (e.g. time-to market, maintenance cost, customer satisfaction ...). Strategic indicators may refine other strategic indicators (forming a graph) and they are defined and followed by some Roles of the company. A strategic indicator may be measured by a KPI and it is related to one or several Factors (e.g. customer satisfaction may be related to usability and robustness factors for a specific company). For additional examples of these concepts, see Annex C.3.

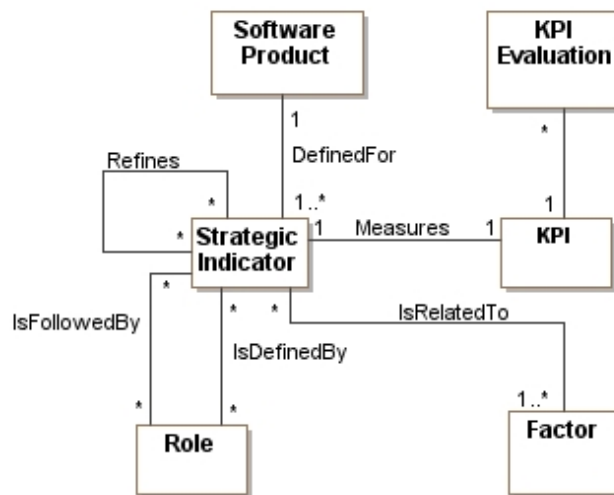


Figure 7: Concepts related to the Strategic Indicator concept

Figure 8 shows the concepts related to the Decision concept. A Decision is a determination arrived at after consideration. Decisions are made by decision-makers, may consider Decision Rules and may be affected by Constraints (either External or Internal). A decision may involve one or more Quality Requirement Actions (see examples of these concepts in Annex C.3).

5.2 Scenarios of Use

In this section, we describe the different scenarios of use of the Dashboard derived from the user stories gathered so far.

- View strategic indicators. The current status of a product in terms of the strategic indicators defined for it will be shown at the beginning of the session (e.g., beginning of the day) and on user demand (user story #63).
- Trend analysis. The trends of the Strategic Indicators in a given period of time will be shown on user demand (#67 and #85).
- Role-dependent view. The information shown (e.g., Strategic Indicators selected, level of detail) will depend on the role played by the user entering in the session (#56 and #72).

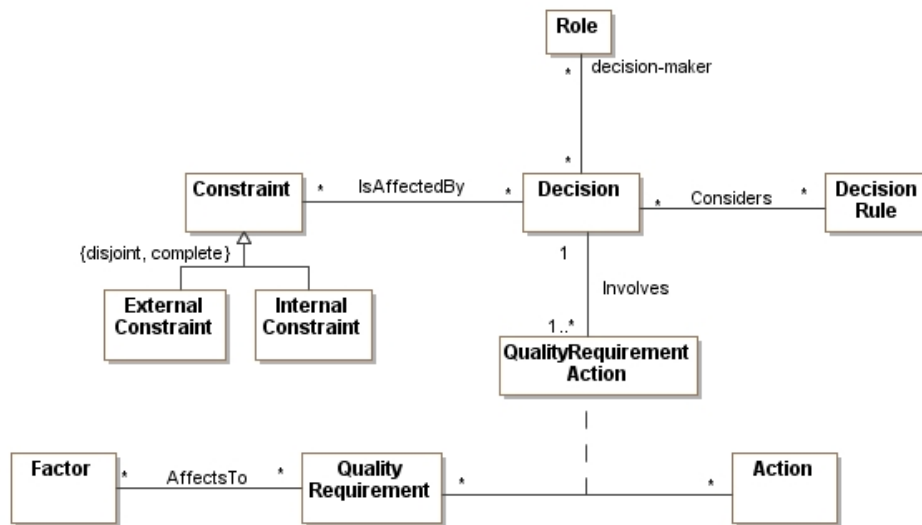


Figure 8: Concepts related to the Decision concept

- Assess a new QR. The dashboard will inform the user when a new QR candidate is identified (provided by the Data Mining engine, see Figure 1). The Strategic Dashboard will support the user in the decision of taking the possible actions for this QR (e.g., include or not this QR in the backlog). This scenario combines the user story related to discovering new QRs (#96), showing new events (#54), the process of supporting a decision (#52, #73) and informing the decision (#55).
- Justify a decision. The Strategic Dashboard should keep all the rationale used when making a decision, to be shown either graphically or textually by generating a report (#55).
- Informing an external constraint. The user need to inform in the Strategic Dashboard when a new external constraint is defined for a product (#94). In a first iteration, we will focus on a concrete external constraint, namely fixing a deadline for a release.
- Informing an internal constraint: limits on the number of simultaneous tasks for a developer. The user need to inform in the Dashboard when new internal constraint is defined (#94). In a first iteration, we will focus on a concrete internal constraint, namely setting the maximum number of tasks in which a developer may be involved simultaneously.

6 User Interface Design

We are using an agile development process, as described in Section 3, following the lean cycle (build, measure and learn) in order to build and re-fine the mock-ups that show the functionalities and serve as a first Proof-of-Concept of the Strategic Dashboard. In this regard, the mock-ups are being used to discuss the different scenarios defined in Section 5.2.

A screenshot of the Strategic Dashboard mock-up is depicted in Figure 8. As shown, the Strategic Dashboard is going to be a web platform since this is requested by the requirement #93. The Dashboard is composed of the following elements:

- **Title:** The brand for the platform.
- **Menu:** A menu containing the top-level navigation items.
- **Login/Logout:** Small section showing the logging information and a logout button.



- **Main screen:** A screen including the main content with all the elements of a topic (e.g. Strategic Indicators) and options to configure the way in which the information is displayed.
- **Notification area:** On top of the main screen, there is a notification area to show any new event that may arise at runtime (e.g. a new Quality Requirement).
- **Auxiliary screens:** A set of customizable auxiliary screens to show some relevant information next to the main screen, such as a calendar with the days until next release, a summary of the status of the backlog, etc.

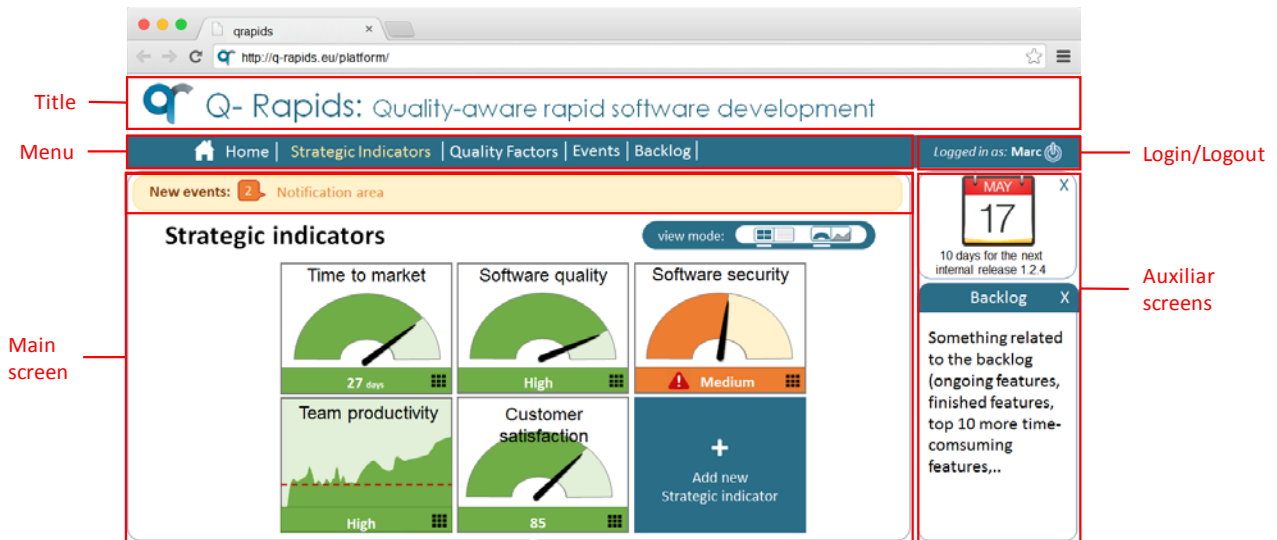


Figure 9: Screenshot of the Strategic Dashboard layout mock-up

We have piloted a first iteration of this lean cycle using the Bittium Use Case. At the current state of development, the mock-ups illustrate two scenarios from Section 5.2:

- **View Strategic Indicators:** The basic case for the dashboard, showing the values of the indicators for the product.
- **Assess a new QR:** The whole process from a new event notifying the arrival of a new QR candidate, to the final decision, passing through the simulation of different scenarios that help the decision-maker.

A complete description and screenshots of the mock-ups in the aforementioned scenarios is shown in Annex D.



Conclusion

This document presents the Strategic Dashboard specification. It includes the overall description of the dashboard, the requirements (functional, quality and non-technical), an ontology to define the concepts that will be used by the dashboard and the conceptual model of the system (class diagram and behavioural description using scenarios). We included, as part of the specification, a first set of mock-ups to be used as departing point in the first development iteration.

As we are following an agile software development process, the requirements (user stories) will be refined in the different development iterations, including new requirements (user stories and/or features) in the backlog. We will use the story stories and mock-ups to include the users (use cases) in the development process.



References

- [Barone et al. 2010] Barone, D.; Mylopoulos, J.; Jiang, L.; Amyot, D (2010). The Business Intelligence Model: Strategic Modelling (Version 1.0). Available at: <ftp://ftp.cs.toronto.edu/csrg-technical-reports/607/BIM-TechReport.pdf>, last access April 2017.
- [Berander-Andrews 2005] Berander, P.; Andrews, A. (2005). Requirements prioritization. In Engineering and managing software requirements, pp. 69-94. Springer Berlin Heidelberg.
- [Ernst et al. 2012] Ernst, N. A.; Murphy, G. C. (2012, September). Case studies in just-in-time requirements analysis. In 2012 Second IEEE International Workshop on Empirical Requirements Engineering (EmpiRE), pp. 25-32. IEEE.
- [Fernández-Lopez et al. 1997] Fernández-López, M.; Gómez-Pérez, A., Juristo, N. (1997). *METHONTOLOGY: From Ontological Art Towards Ontological Engineering*. In: AAAI-97 Spring Symposium Series, 24-26 March 1997, Stanford University, USA.
- [Fitzgerald-Stol 2017] Fitzgerald, B.; Stol, K. J. (2017). Continuous software engineering: A roadmap and agenda. Journal of Systems and Software, Vol. 123, pp. 176-189.
- [Gartner 2017] Gartner Inc. Magic Quadrant for Business Intelligence and Analytics Platforms (February 2017). Doc ID G00301340.
- [Greer-Ruhe 2004] Greer, D.; Ruhe, G. (2004). Software release planning: an evolutionary and iterative approach. Information and software technology, Vol. 46(4), pp.243-253.
- [ISO 9000:2005] International Standardization Organization/International Electrotechnical Commission (2005). 9000: 2005. Quality management systems-Fundamentals and vocabulary
- [ISO 12207:2008] International Standardization Organization/International Electrotechnical Commission (2008). ISO/IEC 12207: 2008. Systems and software engineering– Software life cycle processes.
- [ISO 25010:2011] International Standardization Organization/International Electrotechnical Commission (2011) .ISO/IEC 25010:2011. Systems and software engineering -- Systems and software Quality Requirements and Evaluation (SQuaRE) -- System and software quality models.
- [ISO 26515: 2012] International Standardization Organization/International Electrotechnical Commission (2012). ISO/IEC/IEEE 26515. Systems and software engineering -- Developing user documentation in an agile environment. First edition 2011-12-01; Corrected version 2012-03-15
- [Kaplan et al. 1996] Kaplan, R.S; Norton, D.P; Dorf, RC.; Raitanen. M (1996). In The balanced scorecard: translating strategy into action, Vol 4. Harvard Business School press Boston.



- [Leffingwell, D. 2011] Leffingwell, D. (2011). Agile software requirements: lean requirements practices for teams, programs, and the enterprise. Addison-Wesley Professional.
- [Maté et al. 2014] Maté, A.; Zoumpatianos, K.; Palpanas, T.; Trujillo, J.C.; Mylopoulos, J; Koci, E. (2014). A systematic approach for dynamic targeted monitoring of KPIs. In Proceedings of the 24th Annual International Conference on Computer Science and Software Engineering, pp. 192–206. IBM Corp.
- [OMG-BMM] Object Management Group: Business Motivation Model (BMM) 1.3. (2015). Available at: <http://www.omg.org/spec/BMM/1.3>, last access April 2017.
- [OMG-DMN] Object Management Group: Decision Model and Notation (DMN) 1.1. (2016). Available at: <http://www.omg.org/spec/DMN/1.1/PDF/> , last access April 2017.
- [Palomares et al. 2012] Palomares, C.; Quer, C.; Franch, X.; Guerlain, S.; Renault, S (2012). A catalogue of non-technical Requirement Patterns. In Proceedings of the Second International Workshop on Requirements Patterns (RePa 2012), pp. 1-6.
- [Wagner et al. 2012] Wagner, S.; Lochmann, K; Winter, S.; Deissenboeck, F.; Juergens, E.; Herrmannsdoerfer, M.; Heinemann, L.; Kläs, M.; Tendowicz, A.; Heidrich, J.; Ploesch, R.; Goeb, A.; Koerner, C.; Schoder, K.; Streit, J.; Schubert, C. (2012). The Quamoco quality meta-model, Technical Report TUM-I1281, Technische Universität München.
- [Wallace-Cherniavsky 1990] Wallace, D. R.; Cherniavsky, J. C. (1990). In Guide to software acceptance, Vol. 500, No. 180. DIANE Publishing.



Annex A. Semi-structured interview conducted at Q-Rapids industrial partners' premises

Less than

5
Minutes
Elapsed
 0

1. WARM-UP QUESTIONS:

Q1.1: **Explain your role in the company?** [We expect to interview decision-makers regarding the strategic vision of the company]

- ✓ *What are the decision-making tasks associated to your role in the company?*
- ✓ *How long have you been working in the company?*
- ✓ *How your decisions affect the software products of the company?*
- ✓ *What is your professional background? (e.g., management, informatics....)*
- ✓ *Do you have any experience in software development? If yes how long and in what roles? [Experience in traditional, agile and rapid software development before working in this company]*
 - *How many years of experience in agile and rapid software development?*
 - *In what roles?*

Less than

15
Minutes
Elapsed
 5

2. STRATEGIC GOALS OF THE COMPANY:

Q2.1 Which are the main STRATEGIC GOALS¹¹ of the company?

Q2.2 Which roles define and manage the STRATEGIC GOALS of the company?

- ✓ *How do you interact with these roles?*

Q2.3 How the success of the company STRATEGIC GOALS is measured? (STRATEGIC INDICATORS)

- ✓ *Do you use indicators? Which ones?*
- ✓ *How are these indicators measured?*
- ✓ *How are these indicators related to the Product (MODELIO) goals? Which values indicate the success or failure of the STRATEGIC GOALS of the company?*
- ✓ *Which roles define and manage these indicators?*

3. STRATEGIC GOALS OF THE PRODUCT (Modelio):

Q3.1 Which are the PRODUCT STRATEGIC GOALS?

Q3.2 Which roles define and manage the PRODUCT STRATEGIC GOALS?

- ✓ *How do you interact with these roles?*

Q3.3. How is the success/failure of the software product measured (Modelio product line or a sub-product in the product line)? (PRODUCT INDICATORS)

- ✓ *Do you use some KPIs? Which ones?*
- ✓ *Are these KPI related to the Quality Requirements (QR)?*
- ✓ *How are these KPI (included QR) measured?*
- ✓ *How are they related to the PRODUCT STRATEGIC GOALS? Which values indicates the success or failure of the PRODUCT STRATEGIC GOALS?*

Less than

15
Minutes
Elapsed
 20

4. RELATION BETWEEN STRATEGIC GOALS OF THE COMPANY AND GOALS OF THE PRODUCT (Modelio):

Q4.1 Which STRATEGIC GOALS of the company are related to the PRODUCT STRATEGIC GOALS?

- ✓ *In your opinion, which are the most important STRATEGIC GOALS of the company for the product? Why?*

¹¹ Strategic goals lead decisions in the different level of decision-making processes



- ✓ *How is the relationship among Strategic goals of the company and the goals of the product stated?*

Q4.2 How the STRATEGIC INDICATORS of the company are related to the PRODUCT STRATEGIC GOALS?

- ✓ *How is this relationship stated?*

Q4.3 How the product success (PRODUCT INDICATORS) is related to the STRATEGIC GOALS of the company?

- ✓ *How is this relationship stated?*

Q4.4 How the product success (PRODUCT INDICATORS) is related to the STRATEGIC INDICATORS of the company?

- ✓ *How is this relationship stated?*

Less than

🕒 10

Minutes

Elapsed

🕒 35

5. TOOLS OR PROCESSES CURRENTLY USED IN THE COMPANY

Q5.1 What tools/processes are used for dealing with the company and the Product STRATEGIC GOALS?

Q5.2 What tools/processes are used for dealing with the STRATEGIC INDICATORS of the company and the PRODUCT INDICATORS?

Q5.3 Currently, do you use any dashboard or any other tools for planning, monitoring, and controlling the projects?

If yes

- ✓ *What kind of information is provided/used by the current dashboard/tools? Where the information is coming from? Do you miss some information?*
- ✓ *What are the strengths and drawbacks of the current dashboard/tools?*
- ✓ *Who use these tools?*

Less than

🕒 10

Minutes

Elapsed

🕒 45

6. Q-RAPIDS EXPECTATIONS

Q6.1 Which information do you think that would be valuable (and when) to improve your decision-making process regarding quality requirements (for your software development process and for your company strategic processes)?

- ✓ *How to relate such valuable information with your company strategic decisions?*
- ✓ *How to relate such valuable information with your software development processes?*

Q6.2 Which roles of the company are the target user of the Q-Rapids Dashboard tool?

Q6.3 What are the main expectations of the Q-Rapids Dashboard tool?

- ✓ *Which are the most important aspects (functionalities) that you think that would be required from the expected Q-Rapids Dashboard? Why? (e.g., What-if analysis, other mitigation-strategies used in the company?)*
- ✓ *Which are the roles that would need these aspects?*
- ✓ *What tools used by managers, developers, etc., are envisaged to be connected to the Q-Rapids dashboard (e.g., Sonar)?*
- ✓ *Is there any preference about the user interface design of the Q-Rapids Dashboard?*
- ✓ *What are the main challenges that you think that are related to Q-Rapids Dashboard and tools?*
- ✓ *What type of reports would you like Q-Rapids to provide?*

Less than

🕒 5

Minutes

Elapsed

🕒 55

7. ADDITIONAL COMMENTS:

Q7.1: Are there **any related issues that we missed** and that you would like to reflect on?



Annex B. Strategic Dashboard Requirements

The issue number (characterised by the character #) is the identification given by the tool used to manage the project requirements (Redmine). In this tool, we are managing all the requirements for the whole project. Therefore, this annex only contains a subset of the whole list of the project requirements.

From Table 4 to Table 12 we have the user stories related to each of the themes defined; and Table 13 includes the non-technical requirements.

In the column of the proponent, we anonymise the user case defining the user story using the term UC instead of the real name of the industrial partner.

Table 4 Strategic Dashboard user stories (theme Analysis)

ID	#	Priority	Type	Subject	Proponent
WP3-ANLS-01	58	Normal	functional	As product manager, I want the system to monitor the quality of the product, so that user can know the quality of the product at different times.	UC
WP3-ANLS-02	60	Normal	functional	As product manager, I want the dashboard suggests new features of improvements of existing ones, so that we can anticipate the user's expectations	UC
WP3-ANLS-03	59	Normal	functional	As product owner, I want the system to monitor the team members work (e.g. productivity on Monday), so that we can make better estimations	UC
WP3-ANLS-04	95	Normal	functional	As project manager, I want the Dashboard identifies mitigation activities, so that we can mitigate future risks	UPC
WP3-ANLS-05	97	Normal	functional	As product manager, I want the Dashboard predict violations, so that we can apply mitigation activities before something is wrong	UPC
WP3-ANLS-06	96	Normal	functional	AS product owner, I want Dashboard identify new Quality Requirements to be added to our backlog	UPC

Table 5 Strategic Dashboard user stories (theme Characterisation)

ID	#	Priority	Type	Subject	Proponent
WP3-CHAR-01	57	Normal	functional	As product manager/owner, I want the system to take pictures for different times manually and automatically, so that user can compare current status whit past status.	UPC
WP3-CHAR-02	64	Normal	functional	As product manager, I want a value stream visualization, so that the waste can be reduced	UC
WP3-CHAR-03	70	Normal	functional	As product owner, I want to have milestones related information, so I can see what a milestone is for	UC
WP3-CHAR-04	94	Normal	functional	As product manager, I want the Dashboard store the constraints, so that they can be used to take decisions	UPC
WP3-CHAR-05	88	Normal	functional	As platform integrator, I want to keep hardware dependencies with software	UC



Table 6 Strategic Dashboard user stories (theme Compatibility)

ID	#	Priority	Type	Subject	Proponent
WP3-COMP-01	75	Normal	quality	As a user, I want to have interoperability with other tools (e.g export to MS Word, Adobe,...), so that the results can be more useful	UPC

Table 7 Strategic Dashboard user stories (theme Interaction)

ID	#	Priority	Type	Subject	Proponent
WP3-INTR-01	52	Normal	functional	As product manager, I want setup a hypothesis and see what happens if the parameters change, so that I can take informed decisions	UC
WP3-INTR-02	66	Normal	functional	As product manager/owner, I want to compare the quality of different products, so that we can be better in new products	UC
WP3-INTR-03	73	Normal	functional	As product manager, I want to have an analysis of alternatives when you need to take a decision, so that I can consider different scenarios before taking the decision	UC
WP3-INTR-04	87	Normal	functional	As R&D release manager/Product Manager, I want to have a kind of scenario planning tool	UC

Table 8 Strategic Dashboard user stories (theme Management)

ID	#	Priority	Type	Subject	Proponent
WP3-MNG-01	126	Normal	functional	As a project manager, I want the system manage different roles	UPC
WP3-MNG-02	127	Normal	functional	As a system administrator, I want the system has user management facilities	UPC

Table 9 Strategic Dashboard user stories (theme Reporting)

ID	#	Priority	Type	Subject	Proponent
WP3-REP-01	55	Normal	functional	As product owner, I want to summarize a decision including hypothesis, goals, measurement means and different scenarios analysed, so we will be able to inform them giving visibility to the process behind them.	UC
WP3-REP-02	69	Normal	functional	As product manager, I want to create a Product Plan Report, so that I have every plan of each product in one sheet	UC
WP3-REP-03	92	Normal	functional	As product manager, I want something that helps me to prepare project proposals	UC



Table 10 Strategic Dashboard user stories (theme Security)

ID	#	Priority	Type	Subject	Proponent
WP3-SEC-01	101	Normal	quality	As a user, I want my password won't be stored without encryption in the system	UPC
WP3-SEC-02	99	Normal	quality	As user, I want the system should be accessed using authentication	UPC

Table 11 Strategic Dashboard user stories (theme Usability)

ID	#	Priority	Type	Subject	Proponent
WP3-USA-01	74	Normal	quality	As a user, I want to have a easy-to-use system, so that the learning curve was low	2 UCs, UPC
WP3-USA-02	79	Normal	quality	As IT manager, I want to have an easy-to-configure system, so the users could configure the system themselves	UC
WP3-USA-03	78	Normal	quality	As product manager, I want to have a light way integration in the users processes, so that the new tool cannot be considered as intrusive	UC
WP3-USA-04	93	Normal	quality	As R&D release manager, I want that the Dashboard will be a Webpage	UC
WP3-USA-05	83	Normal	quality	As product manager/owner, I want to have red / yellow / green in terms of general quality, so that the problems are quickly visualized	UC
WP3-USA-06	81	Normal	quality	As product manager, I want to have a system that is accessible to most people, so that the users accept the tool	UC

Table 12 Strategic Dashboard user stories (theme Visualisation)

ID	#	Priority	Type	Subject	Proponent
WP3-VIS-01	51	Normal	functional	As product manager, I want to identify the concepts that change through time (evolution management), so that I can trace the changes in goals	UC
WP3-VIS-02	53	Normal	functional	As product manager, I want to see if the tool would allow to align goals from different area of the Company, so that there is more people involved to have a better result.	UC
WP3-VIS-03	54	Normal	functional	As product manager/owner, I want to show the current status and show events, when they happen	UC
WP3-VIS-04	56	Normal	functional	As product manager, I want to include different viewpoints from user role/expertise, so that we see different information from roles in a different way.	UC
WP3-VIS-05	65	Normal	functional	As product manager, I want a data visualization and Big Data emerging practices.	UC
WP3-VIS-06	63	Normal	functional	As product manager, I want (online) quality visible over the products	UC



WP3-VIS-07	85	Normal	functional	As product owner, I want to analyse trend charts to see the evolution of an specific QR over the time, so that estimations can be produced	UC
WP3-VIS-08	103	Normal	functional	As product manager, I want to visualise the maturity of the product	UC
WP3-VIS-09	71	Normal	functional	As product owner, I want to increase visibility/transparency of progress for developer teams, so that I create a sense of urgency and responsibility	UC
WP3-VIS-10	68	Normal	functional	As product owner, I want to increase visibility of every phase of Software development	UC
WP3-VIS-11	72	Normal	functional	As product manager, I want to include different viewpoints depending on the user role/expertise and the different product life-cycle step, so that different users can see different information	UC
WP3-VIS-12	106	Normal	functional	As product manager, I want to visualise trends (typical values for e.g. 100 features)	UC
WP3-VIS-13	105	Normal	functional	As product manager/High management, I want to visualise Great/business as usual/Disaster	UC
WP3-VIS-14	84	Normal	functional	As product manager, I want to have a matrix visualization of roles/life-cycle steps and the information in the cells, so that all the product information is summarised in one sheet	UC
WP3-VIS-15	102	Normal	functional	As product manager, I want to visualise the quality and the efficiency of process phases	UC
WP3-VIS-16	104	Normal	functional	As product owner, I want to visualise the value stream map for a selected feature with regards to time, quality, money and efficiency	UC
WP3-VIS-17	89	Normal	functional	As platform integrator, I want to have a common backlog, so that we can follow the maturity level of the whole features	UC
WP3-VIS-18	90	Normal	functional	As product manager, I want to see the project progress, so that we can know the finished and unfinished deliverables	UC
WP3-VIS-19	91	Normal	functional	As product manager, I want to see the monitory situation of my product portfolio	UC
WP3-VIS-20	67	Normal	functional	As product manager, I want to generate trend charts for QRs at level of product portfolio, so that we have comparable results over the projects.	UC



Table 13. Strategic Dashboard non-technical requirements

ID	Type	Subject	Proponent
WP3-REL-01	<i>non-technical</i>	<i>The proof-of-concept version should be released by the end of February 2018 (M15)</i>	<i>UPC</i>
WP3-REL-02	<i>non-technical</i>	<i>The consolidated version should be released by the end of November 2018 (M24)</i>	<i>UPC</i>
WP3-REL-03	<i>non-technical</i>	<i>The marketable version should be released by the end of August 2019 (M33)</i>	<i>UPC</i>



Annex C. Ontology

Annex E.1. Concepts of Quality Assessment Package

Aggregation

Description: An Aggregation is used to aggregate values of other measures.

Source(s) (only for concepts coming from an external source): QUAMOCO [Wagner et al. 2012]

Data Source

Description: A data source contains information to calculate metrics related to the software product or process.

Source(s) (only for concepts coming from an external source): QUAMOCO [Wagner et al. 2012]

Example/s: Examples are SonarQube, SVN, or Jenkins

Entity

Description: An entity is used to model a part that a software product and its environment consist of (e.g., code, requirements, architecture, any kind of development artefact and its parts). This also includes resources that are required during the development or use of the product, like, software developers or other stakeholders.

Source(s) (only for concepts coming from an external source): QUAMOCO [Wagner et al. 2012]

Example/s: Examples are "Lines of Code", "Number of Architecture Violations", or "Clone Coverage".

Factor

Description: A factor constitutes a property of either the software product (or part of it) or the software process that is related to the product's quality. A factor is always defined in a way that it is possible to determine the degree to which it is present in the product. To indicate that a factor refers to a certain part of the product, the environment of the product or to a resource, the "characterizes" relationship is used to state an entity.

Source(s) (only for concepts coming from an external source): QUAMOCO [Wagner et al. 2012]

Instrument

Description: An instrument is used to determine the value (of the measure) directly using an external tool or a manual assessment (comparable to a sensor).

Source(s) (only for concepts coming from an external source): QUAMOCO [Wagner et al. 2012]

Metric

Description: A metric defines how a specific attribute of an entity is measured and therefore it provides a means to quantify factors that characterize this entity (or a related one). The measurement can be done by using various techniques (by a manual or tool-based instrument as well as by an aggregation of results of refining measures).

Source(s) (only for concepts coming from an external source): QUAMOCO [Wagner et al. 2012]

Example/s: Examples are "Lines of Code", "Number of Architecture Violations", or "Clone Coverage".

Process Factor

Description: A process factor constitutes a property of the software development process.

Generalizations:

- Factor



Example/s: For example “Productivity”

Product Factor

Description: A process factor constitutes a property of a software product (or part of it).

Generalizations:

- Factor

Example/s: Examples are “Testing” or “Code Quality”.

Quality Factor

Description: A quality factor constitutes a property of the software product

Generalizations:

- Factor

Example/s: Examples are “Reliability” or “Maintainability”.

Annex E.2. Concepts of Quality Aware Rapid Software Development Process Package

Acceptance Criteria

Description: specify the values that a product must meet for acceptance (e.g., a performance requirement that a function must be executed within one second).

Source(s): [Wallace &Cherniavsky: 1990]

Agile Development

Description: software development approach based on iterative development, frequent inspection and adaptation, and incremental deliveries, in which requirements and solutions evolve through collaboration in cross-functional teams and through continuous stakeholder feedback.

Source(s): [ISO 26515: 2012]

Developer

Description: role responsible for writing the code of the story.

Source(s): [Leffingwell, D. 2011]

Epic

Description: Highest level expression of a customer or business need.

Source(s): [Leffingwell, D. 2011]

Feature

Description: functional or non-functional distinguishing characteristic of a system, usually an enhancement to an existing system.

Source(s): [ISO 26515: 2012]

Feature Team

Description: long lived cross functional team that completes many end to end customer features, one by one.

Source(s): [Leffingwell, D. 2011]

Process

Description: a set of interrelated or interacting activities which transforms inputs into outputs.



Source(s): [ISO 9000:2005]

Product Manager

Description: person who is responsible for defining the features of the system at program level.

Source(s): [Leffingwell, D. 2011]

Product Owner

Description: is responsible for determining and prioritizing user requirements, managing the product backlog.

Source(s): [Leffingwell, D. 2011]

Project Portfolio

Description: collection of projects that addresses the strategic objectives of the organization.

Source(s): [ISO 12207: 2008]

Quality

Description: degree to which a set of inherent characteristics fulfils requirements.

Source(s): [ISO 9000: 2005]

Quality Requirement

Description: A quality requirement is a requirement which is not related to functional aspect of software.

Example/s: Improve the user interface of the software product.

Rapid Software Development

Description: evolutionary step from agile software development that focuses on **organizational** capability to develop, release, and learn from software in **rapid parallel cycles**, such as hours, days or very few weeks'

Source(s): [Fitzgerald-Bstol 2017] [Mäntylä et al. 2015]

Release

Description: describes an increment that is valuable to customers and evolves into complete software product

Source(s): [Greer& Ruhe: 2004]

Requirement

Description: need or expectation that is stated, generally implied or obligatory.

Source(s): [ISO 9000: 2005]

Requirements prioritization

Description: Crucial and integral part of software decision making that helps to identify the most valuable requirements among candidate requirements that need to be realized within time and cost constraints.

Source(s): [Berander-Andrews 2005]

Sprint

Description: short time frame, in which a set of software features is developed, leading to a working product that can be demonstrated to stakeholders. NOTE In some organizations, a sprint is known as an iteration.

Source(s): [ISO 26515: 2012]

Task



Description: Task is an individual work item which compose a requirement, are commonly assigned to individuals, and in many cases are formally represented in issue trackers and code repositories.

Source(s): [Ernst et al.:2012]

Team backlog

Description: typically called project or product backlog consists of all the user stories the team has identified for implementation.

Source(s): [Leffingwell, D. 2011]

Tester

Description: integral part of agile team responsible for writing acceptance test case while the code is written and testing it against the acceptance criteria.

Source(s): [Leffingwell, D. 2011]

User Story

Description: simple narrative illustrating the user goals that a software function will satisfy.

Source(s): [ISO 26515: 2012]

Annex E.3 Concepts of Strategic Decision Making Package

Action

Description: An action is a process of doing something, typically to achieve an aim.

Example/s: Move a quality requirement to the backlog.

Attributes:

- *name:* String. Action name
- *description:* String. Action description

Associations:

- *qualityRequirementAction:* QualityRequirementAction [*]. References the quality requirements which the action has been applied.

Constraint

Description: A constraint is a condition or restriction that affects to the strategic decisions.

Example/s: Developers cannot work in more than two tasks during the same week.

Attributes:

- *description:* String. Constraint description.

Associations:

- *decisionAffected:* Decision [*]. References decisions affected by the constraint.

Decision

Description: A Decision is a determination arrived at after consideration.

Example/s: The evaluation of the KPI (user stories delivered on-time/user stories planned to be delivered)*100 show a violation since its value is 40% when the threshold indicates that the value must be greater than 95%. In the light of this value the project manager makes the decision of hiring another developer.

Attributes:

- *description:* String. Decision description.
- *time:* Time



Associations:

- **qualityRequirementAction:** QualityRequirementAction [1..*]. References the actions to perform over the quality requirements involved in the decision made.
- **decision-maker:** Role [*]. References the roles that made the decision.
- **decisionRule:** DecisionRule [*]. References the decision rules considered when the decision is made.
- **constraint:** Constraint [*]. References the constraints that affect to the decision.

Decision-Maker

Description: A decision-maker is a person who makes decisions on quality requirements in the software development process.

Example/s: Product manager, Product owner

Decision Rule

Description: A decision rule is a rule that encode preferences of decision-makers when several decision alternatives are available or conflicts rise.

Example/s: If a conflict arises in a decision between quality levels and time to market, quality levels will be prioritized.

Attributes:

- **rule:** String

Associations:

- **decision:** Decision [*]. References decisions that should consider the decision rule.

External Constraint

Description: An external constraint is a constraint that represents conditions that are out of the control of decision-makers.

Example/s: Project budget as approved by management or a particular event date on which a robust release wants to be demonstrated.

Generalizations:

- Constraint

Factor

Description: A factor constitutes a property of the software product (or part of it) that is related to the product's quality. A factor is always defined in a way that it is possible to determine the degree to which it is present in the product. To indicate that a factor refers to a certain part of the product, the environment of the product or to a resource, the “characterizes” relationship is used to state an entity.

Source: Quality Assessment Package

Example/s: Redundancy, maintainability, etc...

Attributes:

- **name:** String. Factor name.

Associations:

- **relatedStrategicIndicator:** StrategicIndicator [*]. References the strategic indicators related to the factor.



Internal Constraint

Description: An internal constraint is a constraint that encodes conditions on the development process that may eventually influence decision-making.

Example/s: Developers cannot work in more than two tasks during the same week.

Generalizations:

- Constraint

KPI

Description: A KPI is a metric that measures the degree of achievement of a strategic indicator. This concept is based on the Indicator concept defined in BIM language [Barone et al. 2010].

Example/s: (user stories delivered on-time / user stories planned to be delivered)*100.

Attributes:

- *name*: String. KPI name.
- *description*: String [0..1]. KPI description.
- *unitOfMeasure*: Unit
- *expression*: Expression
- *target*: Float
- *upperThreshold*: Float [0..1]
- *lowerThreshold*: Float [0..1]

Associations:

- *strategicIndicatorMeasured*: Strategic Indicator [1]. References the strategic indicator to be measured by the KPI.
- *kpiEvaluation*: KPIEvaluation [*]. References the measurements of the KPI at several points of time.

Constraints:

- One of the thresholds has to have a value.
- The value of the target must be between the lowerThreshold and upperThreshold.

KPI Evaluation

Description: A KPI Evaluation is a measurement of a KPI at a certain point of time.

Example/s: If the number of user stories delivered on-time at the end of an iteration of the project is 2 and the number of user stories planned to be delivered is 6, the evaluation of the KPI (user stories delivered on-time/user stories planned to be delivered)*100 is 40%.

Attributes:

- *currentValue*: Float
- *evaluationTime*: Time

Associations:

- *kpi*: KPI. References the KPI used for the KPI evaluation.

Quality Requirement

Description: A quality requirement is a requirement which is not related to functional aspect of software.

Example/s: Improve the user interface of the software product.

Source: Quality Aware Rapid Software Development Process Package

Attributes:

- *description*: String. Quality requirement description

Associations:



- *qualityRequirementAction*: QualityRequirementAction [*]. References the actions which have been applied to the quality requirement.
- *factor*: Factor [*]. References the factors to which the quality requirement affects to.

Quality Requirement Action

- **Description**: A quality requirement action is a relationship between a quality requirement and an action applied to it.
- **Example/s**: Move a quality requirement improve the user interface to the backlog.

Role

Description: The position or purpose that someone has in the organization.

Example/s: Product manager, developer,...

Attributes:

- *name*: String. Name of the role.
- *description*: String [0..1]. Role explanation.

Associations:

- *definedStrategicIndicator*: Strategic Indicator [*]. References the strategic indicators defined by the role.
- *followedStrategicIndicator*: Strategic Indicator [*]. References the strategic indicators that are followed by the role.

Software Product

Description: A software product is a computer program designed to perform a group of coordinated functions, tasks, or activities for the benefit of the user.

Example/s: A modelling tool is a software product.

Attributes:

- *name*: String. Software product name.

Associations:

- *strategicIndicator*: Strategic Indicator [1..*]. References the strategic indicators defined for the software product.
- *composedBy*: Entity [*]. References the entities composing the software product.

Strategic Indicator

Description: A Strategic Indicator is an aspect that a company considers relevant for the decision making process during the software process development.

Example/s: Time-to-market, maintenance cost, customer satisfaction.

Attributes:

- *description*: String. Strategic Indicator explanation.

Associations:

- *softwareProduct*: SoftwareProduct [1]. References the software product for which the strategic indicator is defined.
- *measure*: KPI [0..1]. References the KPIs that measure the strategic indicator.
- *refinedStrategicIndicator*: Strategic Indicator [*]. References the strategic indicators that the strategic indicator depends on in some way.
- *factor*: Factor [1..*]. References the factors the strategic indicator is related to.



-
- *roleInChargeToDefine*: Role [*]. References the roles that define the strategic indicator.
 - *roleInChargeToFollow*: Role [*]. References the roles that may follow the progress of the strategic indicators.



Annex D. Mock-ups

In this Annex, we depict the screenshots of the Mock-ups that show the Strategic Dashboard in the scenarios *View Strategic Indicators* and *Assess a new Quality Requirement*.

View Strategic Indicators

The user can access the screen of Strategic Indicators by clicking on the *Strategic Indicators* item from the Menu. The Strategic indicators can be shown in two different *view modes*:

- Grid view: shows the Strategic Indicators graphically in a grid (see Figure 9).
- Tabular view: shows the Strategic Indicators in a tabular form (see Figure 10).

In any of the two views, the user can see the current values of the Strategic Indicators, and if these values satisfy the objectives by means of some thresholds defined. If a Strategic Indicator does not meet the threshold, the Strategic Indicator is highlighted with an alert mark.

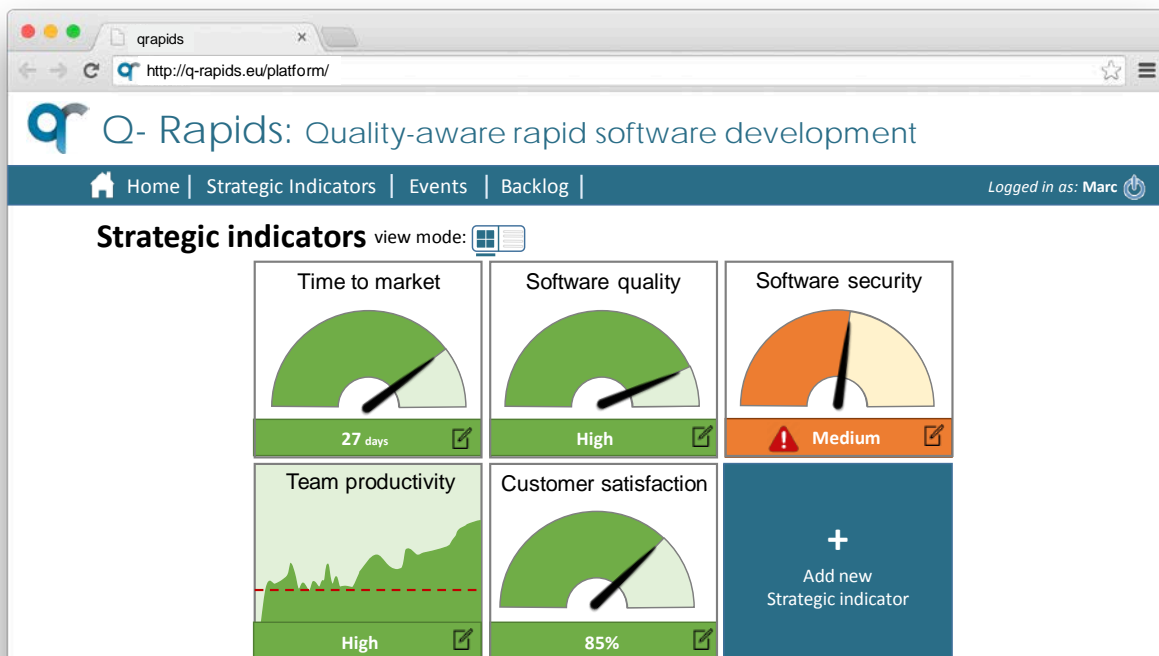


Figure 10: Screenshot of “View Strategic Indicators” in grid view

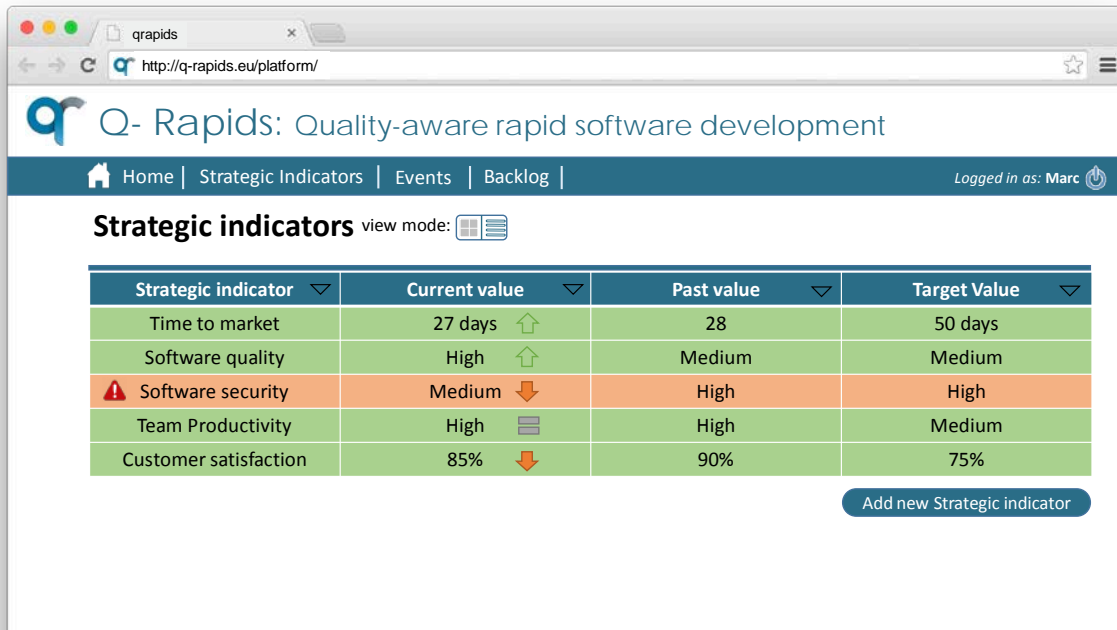


Figure 11: Screenshot of “View Strategic Indicators” in tabular view

Assess a new Quality Requirement

When new Quality Requirement arrives to the Dashboard, it is displayed as a *New Event* in the *Notification area*, located on top of the main screen (see Figure 11). In this example, the notification shows that there are 2 new quality requirements.

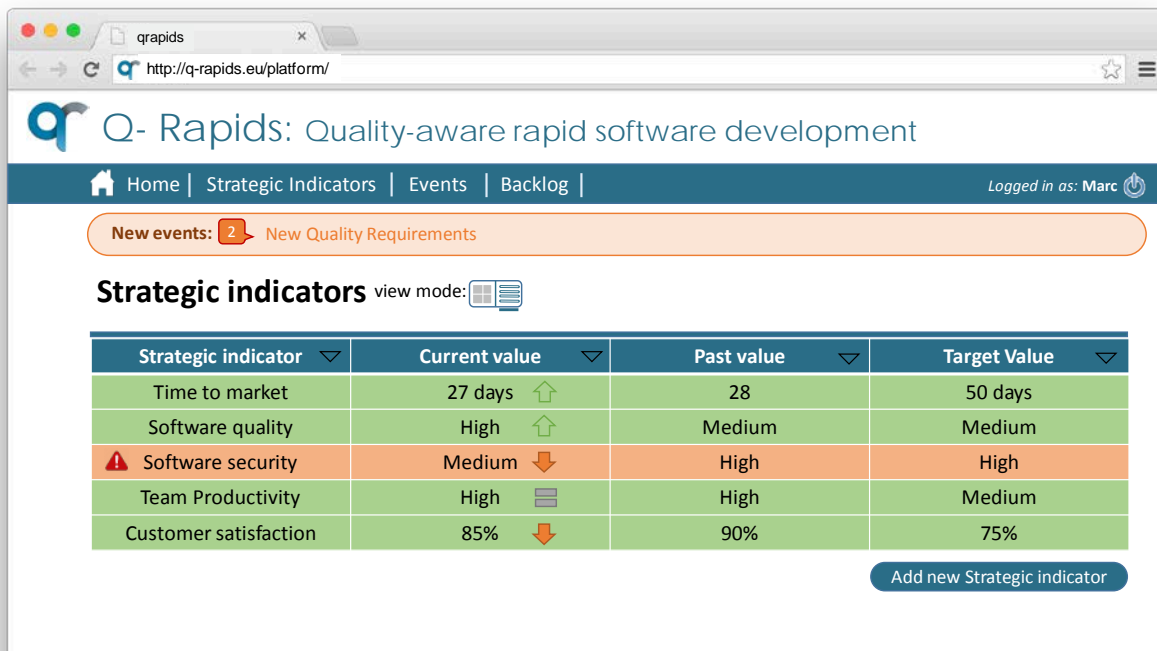


Figure 12: Screenshot of “New Event” notification



When the user clicks on the notification, the system shows the new quality requirements. The quality requirements are displayed with some information, such as a short description, the creation date, etc. It also shows the actions that can be done with each quality requirement (see Figure 12).

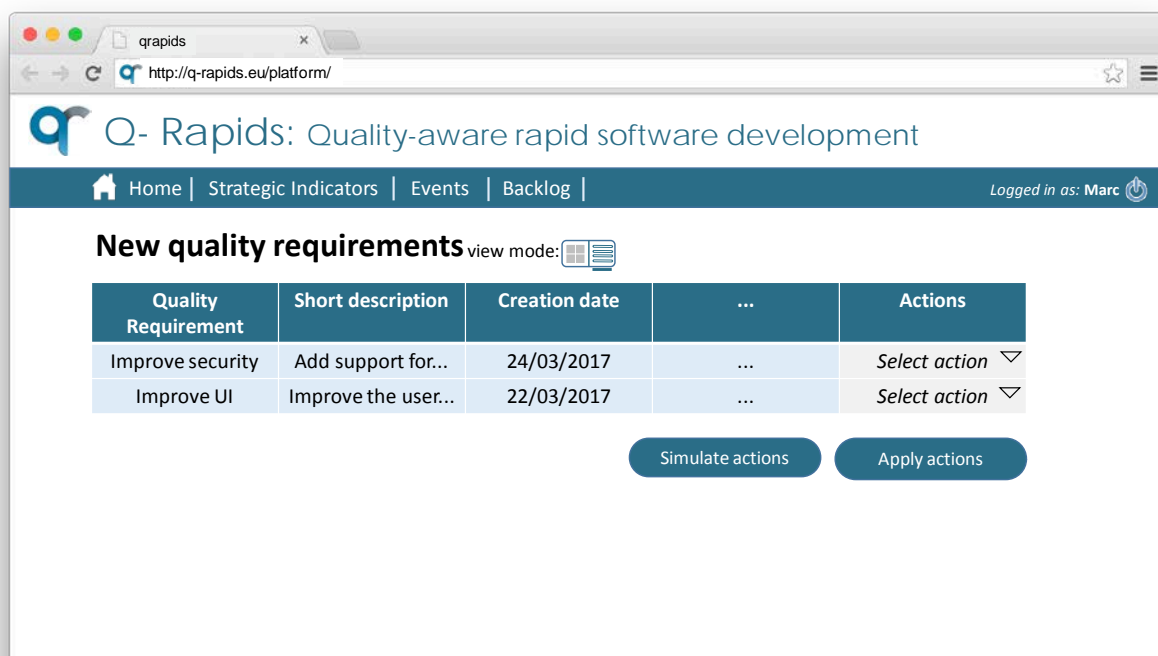


Figure 13: Screenshot of "New quality requirements"

The actions that can be taken are still under discussion, but some tentative options could be to include the quality requirement to the backlog, or ignore the quality requirement (see Figure 13).

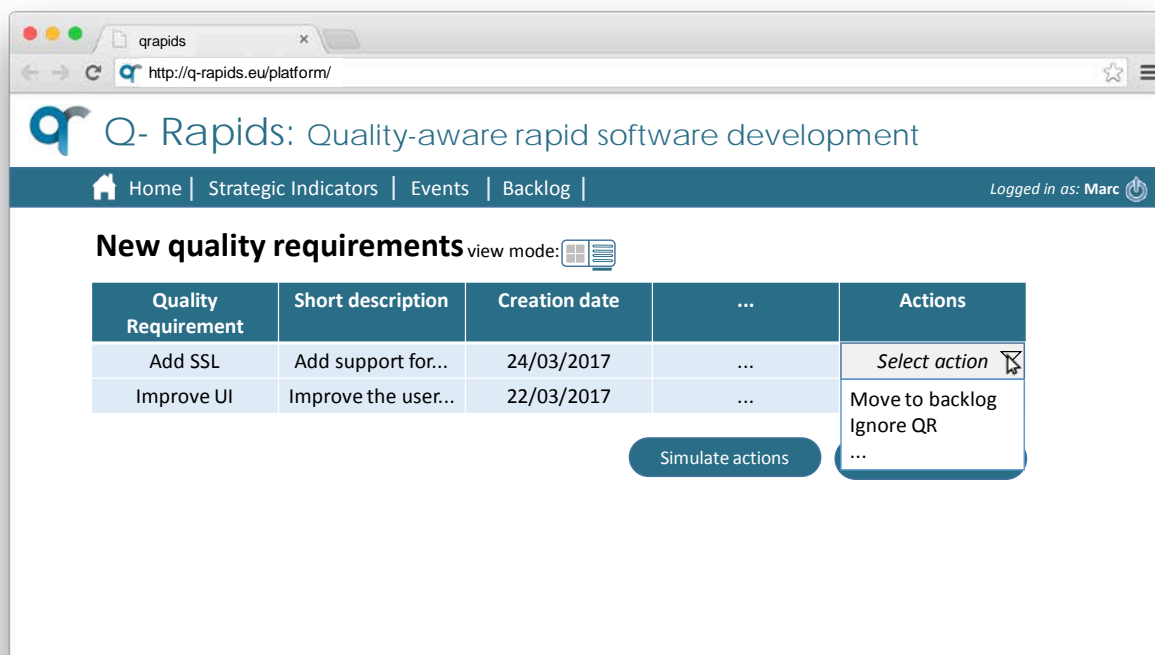


Figure 14: Screenshot of Taking actions on Quality Requirements

The actions taken have an effect to the Strategic Indicators, therefore, some simulation can be conducted in order to help the user in the decision making process. The simulations apply to a set of actions (e.g. put in the backlog some quality requirements and ignore some other). The information and results of the simulations are still under discussion. In any case, after the simulation the user may decide to *Apply* or *Cancel* the selected actions. When the user decides to apply the actions, a pop-up will emerge to let the user introduce a comment to describe or justify the decisions he has made (see Figure 14). This information will later be useful to understand the list of actions done along time.

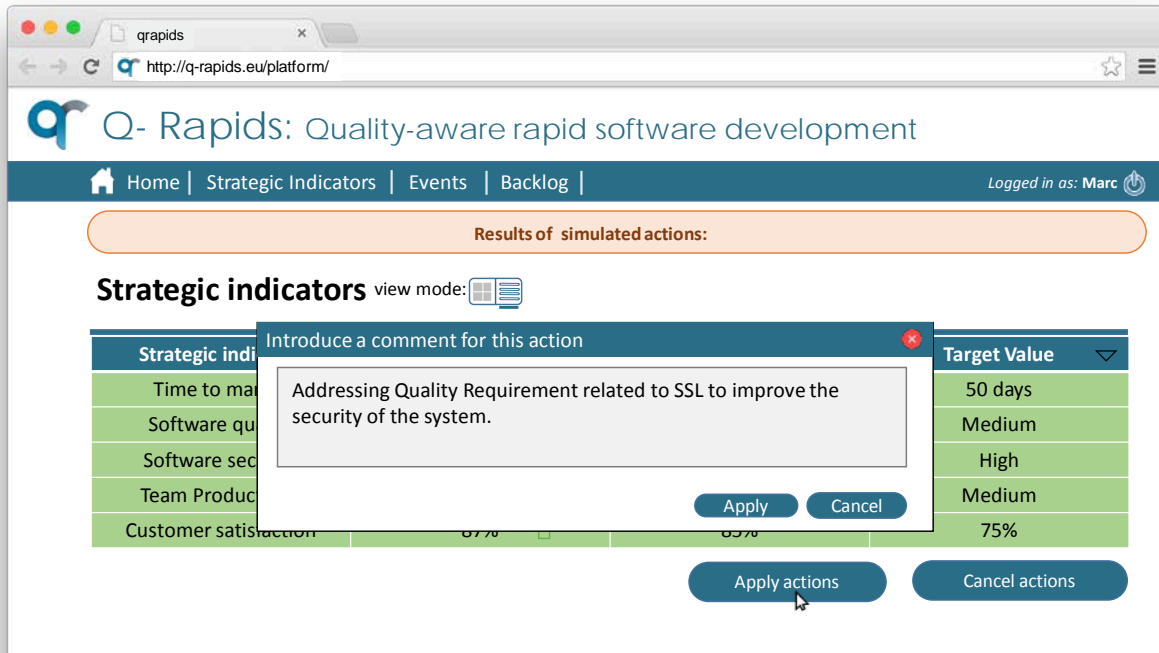


Figure 15: Screenshot of Apply actions