This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 732253.

D2.2 Baseline process metrics: "Metrics to improve"

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	1st November 2016 – 31st October 2019		
Project website	www.q-rapids.eu		
Project WP	WP2 – *Quality-aware Rapid Software Development Process*		
Project Task	Task *2.2 – Process baseline data metrics gathering and analysis*		
Deliverable type	X	R	Document, report
		DEM	Demonstrator, pilot, prototype
		DEC	Websites, patent fillings, videos, etc.
		OTHER	Material that does not belong to any specified category
		ETHICS	Ethics requirement
Contractual delivery	23/10/2018		
Delivered	31/01/2018		
Responsible beneficiary	University of Oulu (OULU)		
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)

Copyright © Q-Rapids consortium – All rights reserved

Version history			
Version no.	Date	Description	Author
V0.1	2/12/2017	Outline and initial content of Sections 1 and 3	Pilar Rodríguez, UOULU
V0.2	15/12/2017	Structure changed based on Markku's feedback	Pilar Rodríguez, UOULU
V0.3	17/01/2018	Updating the document with process metrics, baseline values, and other necessary insights to make it delivery-ready	Prabhat Ram, UOULU
V0.35	17/01/2018	General revision	Pilar Rodríguez, UOULU
V0.4	18/01/2018	Incorporation of feedback from V0.35, and further updates	Prabhat Ram, UOULU
V0.45	19/01/2018	General revision	Pilar Rodríguez, UOULU
V0.5	19/01/2018	Incorporation of feedback from V0.45, and further updates	Prabhat Ram, UOULU
V0.6	23/01/2018	Incorporation of feedback from Internal Reviewer #1	Prabhat Ram, UOULU
V0.7	24/01/2018	Incorporation of follow-up feedback from Internal Reviewer #1	Prabhat Ram, UOULU
V0.8	26/01/2018	Incorporation of feedback from Internal Reviewer #2	Prabhat Ram, UOULU
V0.9	29/01/2018	Incorporation of feedback from the UC providers	Prabhat Ram, UOULU
V0.95	30/01/2018	General revision	Pilar Rodríguez, UOULU
V1.0	31/01/2018	Incorporating feedback from the "General revision"	Prabaht Ram, UOULU

Copyright © Q-Rapids consortium – All rights reserved

Authors	
Organisation	Name
UOULU	Pilar Rodríguez
UOULU	Prabhat Ram
UOULU	Markku Oivo
Reviewers	
Organisation	First name last name
UPC	Lidia López
ITTI	Michal Choras

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.

Copyright © Q-Rapids consortium – All rights reserved

Definition of the key terms and abbreviations	
A term or an abbreviation (alphabetical order)	Explanation of the term or the abbreviation
D	Deliverable
DoA	Description of Action
GQM	Goal- Question-Metric
MX	Month X (e.g. M15, Month 15)
PO	Project officer
PoC	Proof-of-Concept
PTSW	Production Testing SoftWare
QM	Quality Model
QR	Quality Requirement
SVN	Apache Subversion
T	Task
UC	Use Case

Copyright © Q-Rapids consortium – All rights reserved

Contents

Copyright © Q-Rapids consortium – All rights reserved

Copyright © Q-Rapids consortium – All rights reserved

The list of tables

Copyright © Q-Rapids consortium – All rights reserved

The list of figures

Copyright © Q-Rapids consortium – All rights reserved

Executive summary

This deliverable provides a measurement plan for analyzing Q-Rapids process performance and quality as part of Task (T) 2.2 '*Process baseline data metrics gathering and analysis*'. Particularly, it presents the set of metrics to be used in T2.2 and the measurement plan to operationalize those metrics. The metrics presented in this deliverable will allow to objectively assess the impact that Q-Rapids has on process performance and quality. The goal is to compare Use Cases (UCs) lifecycle processes' efficiency and quality before and after incorporating Q-Rapids solutions by using these metrics. In addition, these metrics will create objective basis for guiding process improvement actions within UCs ("Metrics to improve").

To create a reliable set of metrics that are useful for both the Q-Rapids project and specific UCs, we conducted Goal-Question-Metric (GQM) workshops with all UC providers. Metrics elicited during the workshops were aggregated and consolidated as presented in this deliverable. Strategic indicators, factors (both product and process), and metrics considered in the Q-Rapids Quality Model (QM) have been considered during the GQM workshops to analyze their relevance for T2.2. This deliverable contains:

- A measurement plan for T2.2, including the set of metrics to be used, a plan for collecting data based on these metrics, a plan for analyzing collected data and comparing results in order to understand the impact of the Q-Rapids software development process and a plan for visualizing results.
- A first baseline for the metrics in each Q-Rapids' UC collected in the Month 15 (Baseline M15).

The content of this deliverable will be used by T2.3 and T2.4 to assess the impact of the transformation towards Q-Rapids processes, principally from the point of view of processes for managing quality requirements (QRs). Deliverable 2.2 will be updated as process metrics may evolve along the project as a result of its usage.

Copyright © Q-Rapids consortium – All rights reserved

1. Introduction

This document presents the baseline process metrics - "Metrics to improve" that are considered as part of T2.2. Particularly, the document presents:

- A measurement plan that defines the set of metrics to be considered in the scope of T2.2, how metrics will be collected, their values stored, analyzed, and visualized.
- Baseline values for each metric in each UC (current values, before introducing Q-Rapids solutions).

In addition, this document describes the way in which these metrics have been elicited by conducting GQM workshops with each UC.

1.1 Motivation

There are twofold motivations for defining baseline process metrics – "Metrics to improve", which are as follows:

1. To analyze the effects of Q-Rapids solutions in software development process performance and quality (comparing software development process before and after using Q-Rapids solutions).
2. To guide process improvement opportunities in each Q-Rapids' UC (i.e. identification of software development process areas that can be improved in each individual UC).

1.2 Intended audience

The intended audience is the Q-Rapids project officer (PO), members, including reviewers and participating organizations and general audience outside the Q-Rapids project:

- Q-Rapids members, including both UC's providers and researchers, will use the metrics and measurement plan defined in this document. UC's providers will mainly use these metrics to guide their software process improvement opportunities. Researchers will use these metrics to analyse the effects of Q-Rapids solutions in software processes.
- Reviewers and PO will peruse the reported effects of Q-Rapid solution's introduction into UC's software development processes.
- As this specification is a public document, it will be accessible by any person interested on "Metrics to Improve" in the context of rapid software development. The main beneficiaries of the deliverable outside the consortium are the organizations or companies that may be willing to use Q-Rapids type approach in their software development processes.

1.3 Scope

The scope of this document is the entire Q-Rapids project. The current document is related to the activities scoped under "T2.2 Process baseline data metrics gathering and analysis". Based on this task, this document elaborates upon the process of collecting the baseline process metrics. Specifically, the document reports the knowledge gathered, exchanged, and actionable data collected in the process of collecting Baseline process metrics: "Metrics to improve".

1.4 Relation to other deliverables

This deliverable is related to the following past and contemporary deliverables. In addition, the outcomes of this deliverable will also influence deliverables D2.3 – D2.6.

- *D2.1: Baseline process definition: "State of the practice"* – D2.2 extends upon this past deliverable by establishing the set of metrics to operationalize and quantify improvement areas identified therein.

Copyright © Q-Rapids consortium – All rights reserved

- *D1.2: Data gathering and analysis proof-of-concept* and *D3.2: Dashboard proof-of-concept* – Some of the baseline process metrics reported in D2.2 stem from the Q-Rapids QM that was presented in the GQM workshops. A subset of this QM has been operationalized for the PoC in D1.2. Some of these metrics have been visualized through the Q-Rapids dashboard, which is reported in D3.2.

In addition to the above deliverables, outcomes of D2.2 will also have a bearing on future deliverables D2.3 – D2.5 for the following reasons:

- *D2.3, D2.4: Construction and validation of the Q-Rapids process* – Baseline process metrics collected in D2.2 will help evaluate the impact of the Q-Rapids solution. Through an iterative process of evaluation and integration, the baseline process metrics will help guide Q-Rapids solution development further.
- *D2.5: Q-Rapids process description* – Baseline process metrics will be used as a standard against which Q-Rapids process performance and quality will be compared to validate the solution developed throughout D2.3 and D2.4.

1.5 Structure of the deliverable

This deliverable is organized into the following sections: Section 1 introduces the document. Section 2 describes the objective of T2.2 and how GQM has been used as a means to achieving those objectives. Section 3 presents the measurement plan, including the set of process metrics that will be used to analyze Q-Rapids process performance and quality, and the measurement plan that will be used to operationalize these metrics in each UC, including measurement mechanisms, data storage, data analysis and visualization. Section 4 presents baseline values for these metrics in each UCs. Conclusions are presented at the end of the document.

Copyright © Q-Rapids consortium – All rights reserved

2. T2.2 Objective and GQM as a means to achieve that objective

This section describes the objective pertaining to *T2.2*, viz. *Process baseline data metrics gathering and analysis*. The subsequent subsections highlight and decompose this singular objective into multiple manageable sub-objectives that have been attained in the course of this task.

2.1 T2.2 Objective

According to the DoA, the objective of T2.2 is as follows:

> "T2.2 provides the baseline metrics for analysing Q-Rapid process performance that will be measured and visualized through the Q-Rapids dashboard (WP3) and integrated in the Q-Rapids platform (WP4). The measurements will provide a reliable baseline for comparing current lifecycle process efficiency and quality before and after Q-Rapids' solutions. This task includes the analysis of key quality and performance indicators in rapid software development and throughput times for QRs".

Thus, the main objective of T2.2 is to define a reliable set of metrics to analyse Q-Rapids process performance and quality. This main objective can be decomposed in the following sub-objectives:

- Sub-objective 1: Identify set of metrics that is relevant to each UCs with respect to the improvement areas concerning Q-Rapids process performance and process quality in their respective software development environments, and can be used to monitor the Q-Rapids software development process.
- Sub-objective 2: Identify and aggregate metrics that appear to be common across UCs. Furthermore, identify set of metrics exclusive to each UCs (case-specific metrics).
- Sub-objective 3: Create a baseline for the identified metrics, characterizing the status quo of the UCs prior to official adoption of the Q-Rapid solution. This also helps create a reference point for later assessments of variations in these values and in values of metrics to be collected further.

2.2 T2.2 Overview

Work under T2.2 is a co-creation process between researchers from UOULU and practitioners (UC providers) in which the elicitation of metrics to provide a reliable baseline for comparing software development lifecycle processes before and after using Q-Rapids solutions has been carried out using the GQM method. The goal was eliciting metrics that will not only enable comparing software development processes before and after using Q-Rapids solutions (as stated in T2.2) but also be useful for UC providers to support their process improvement activities (e.g. the metrics will guide the identification of process improvement opportunities). Based on the Quality Improvement Paradigm [2], Figure 1 presents the activities (steps) that have been considered to execute T2.2. Steps on the right side of the figure (characterize, set goals and define metrics, and choose process and tools) are the objects of this deliverable. They will serve the purposes of T2.3 and T2.4, as shown in the left side of the figure.

Copyright © Q-Rapids consortium – All rights reserved

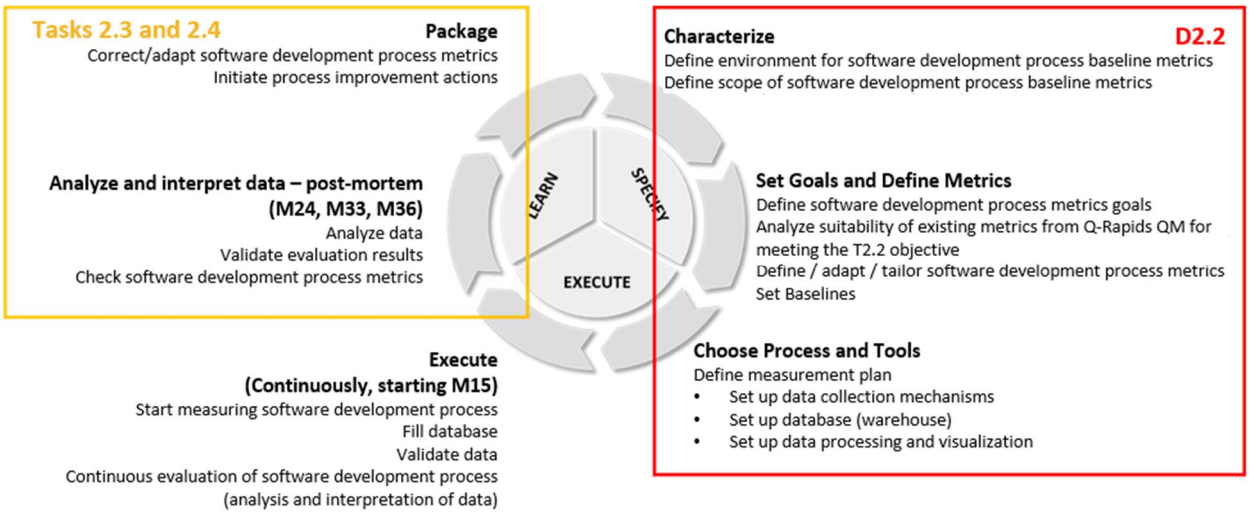

Figure 1. T2.2 overview.

1. The first task consisted of defining the scope in which process baseline metrics will be used in each UC (i.e. scope in which Q-Rapids solutions will be used and, therefore, performance and quality of software development process before and after using Q-Rapids solutions can be measured). This step is important so as to ensure that data is collected from and analyzed in the right environment.
2. Once the scope was defined, measurement goals were set up and metrics to measure those goals were defined. The GQM approach was used in this step. In addition, baseline (i.e. reference) values for those metrics were established.
3. Once the metrics were defined, we worked on defining a measurement plan to establish the data collection mechanisms that will be applied to measure as well as strategies to store collected data, analyse it and visualize results.

A continuous evaluation of the software development process (analysis and interpretation of data) will be carried out in each UC for internal purposes (e.g. finding improvement opportunities in the software development process). At Q-Rapids project level, data will be analyzed and interpreted between M16 and M33 in order to assess the impact of the transformation from the point of view of the processes for managing QRs (T2.3). Moreover, the metrics defined in this deliverable will serve as basis to the empirical validation of best practices, tools and methods defined by T2.3 and integrated in T2.4 between M25 and M36. The overall goal will be to assess the impact of Q-Rapids solutions in UCs' software development processes. These analyses will be reported in deliverables D2.3, D2.4, D2.5 and D2.6 spread across M24, M33 and M36.

2.3 GQM

We applied the GQM method [1], which is based on providing goal-oriented metrics, in order to define process baseline metrics. GQM helps avoid measurement problems such as collecting unnecessary data, and postulating standard measures without adaptation for the environment. This results in useless insights, collecting data that are never analyzed, analyzed in the wrong environment, or missing data, resulting in important aspects not being analyzed. These problems can easily lead to drawing wrong conclusions or resulting in an insufficient pay-off for the cost of the measurement program.

Figure 2 depicts the structure of GQM. In GQM, elicitation of metrics is guided by a measurement goal so as to collect only those metrics that are required by the goal, and nothing extra (exactly the right set of metrics). A set of questions operationally defines the GQM Goal. A set of metrics is associated with every question so as to answer it in a quantitative way. The values to these metrics quantitatively construct the measurement goal, which in turn help UC providers to monitor software development processes to improve them.

Copyright © Q-Rapids consortium – All rights reserved

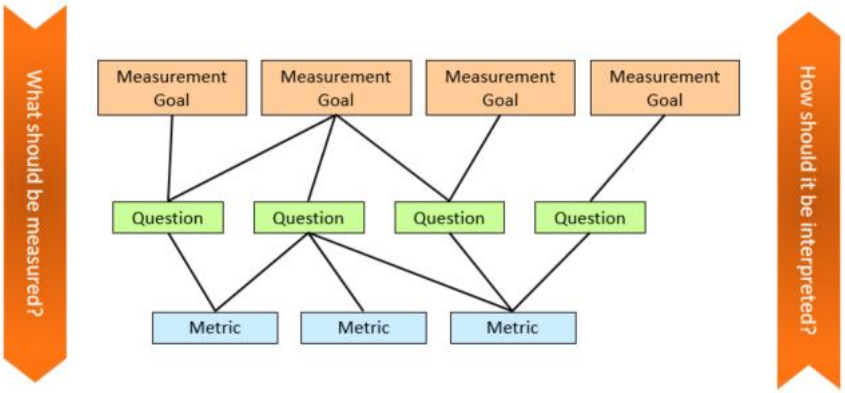

Figure 2. GQM Structure.

The bottom-up approach to interpreting the metric values is essential to ensuring data collected measures the metrics, which in turn answers the questions that decompose measurement goals identified by the UC providers.

2.4 GQM with existing metrics

During the GQM workshops, we aimed to reuse as much as possible (and always when it made sense) factors and metrics already considered in the context of the Q-Rapids project. For this reason, the Q-Rapids QM existing at the beginning of T2.2 was presented to the UC providers during the first GQM workshop in order to evaluate its suitability to assess Q-Rapids process quality and performance (i.e. to check whether the metrics/factors/strategic indicators already considered in Q-Rapids QM would be suitable to meet T2.2's objectives). Figure 3 describes our strategy of using GQM with existing metrics.

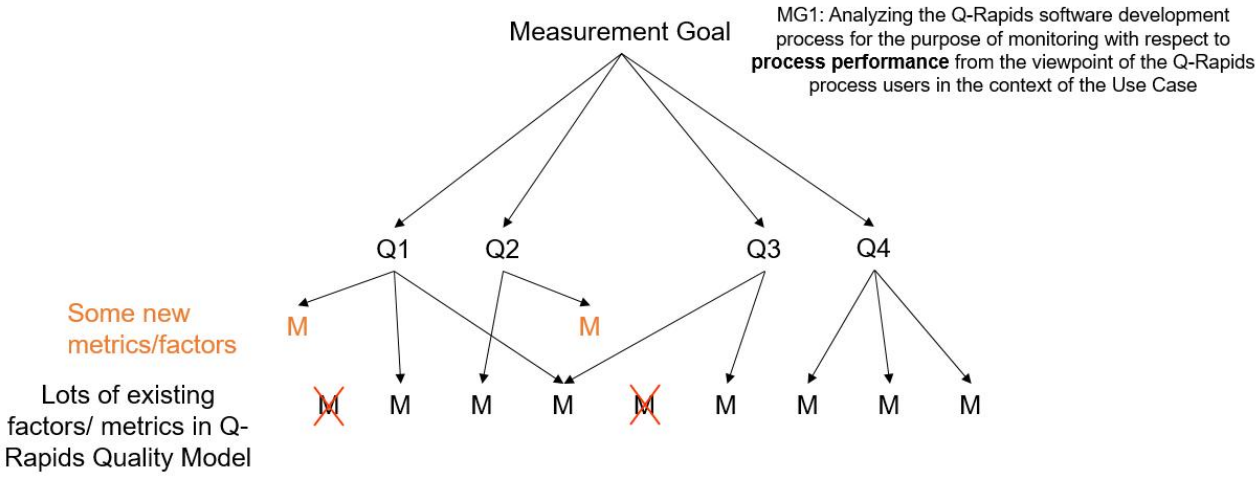

Figure 3. Using GQM with existing metrics.

We explored the suitability of existing product/process factors and metrics already defined as part of the Q-Rapids QM (existing Q-Rapids QM when starting T2.2) to meet T2.2 objectives (i.e. to monitor Q-Rapids software development process performance and quality). UC providers selected those that could be reused and discarded those that were seen as useless in such a context. In addition, new factors/metrics emerged during the GQM workshop, which were not considered in Q-Rapids QM, but were seen as useful by UC providers to measure Q-Rapids process performance and quality and, therefore, are reported in this deliverable. New elicited metrics will be included in following versions of the Q-Rapids QM.

Copyright © Q-Rapids consortium – All rights reserved

Table 1 lists the several existing factors and metrics that were already considered in Q-Rapids QM at the starting of T2.2. A detailed description of these metrics is available in Appendix A. The presented QM is composed of the following components:

1. Strategic Indicator - An aspect that a company considers relevant for the decision-making process. Strategic Indicators are composed of *Product and/or Process Factors*.
2. Product/Process Factors – Product/process factors are attributes of parts of the product/process. They need to be concrete enough to be measured. Product/process factors are composed of *metrics.*
3. Metric – A metric is a concrete description of how a specific product/process factor should be quantified for a specific context. Assessed metrics are calculated from data coming from *data sources.*
4. Raw Data from Data Sources: Raw data are the data as it comes from the different data sources (without any modification). Typically, it cannot be broken down into simpler or more granular forms of data.

Table 1 Existing Q-Rapids Quality Model used during first GQM workshop

Strategic Indicator	Product/Process Factor	Metric
Product Quality, Blocking	Code Quality	Complexity
		Files under the threshold of complexity (%)
		Cognitive Complexity
		Files under the threshold of cognitive complexity (%)
		Files exceeding the comments percentage threshold (%)
		Files under the duplicated lines percentage threshold (%)
		Issues (violations)
		Technical Debt
		Lines of code
		Rules Compliance Index (RCI)
	Testing Status	Condition coverage
		Unit tests violating the condition coverage threshold (%)
		Line coverage
		Unit tests violating the line coverage threshold (%)
		No. of Projects (old jobs)
		Projects passed (%)
		No. of Builds
		Testing stability of a project
		No. of Unit tests
		No. of Unit test errors
		No. of Unit test failures
		Unit test success density (%)
Product Quality, Customer Satisfaction	Stability	No. of bugs and errors in the current sprint
		Customer satisfaction
Blocking	Testing Performance	Unit tests duration
		Tests under the threshold of duration (%)

Copyright © Q-Rapids consortium – All rights reserved

On-time Delivery	Issues' Velocity	Effort needed for unresolved issues with due date in this sprint / Remaining effort in the sprint
		No. of unresolved issues with due date in this sprint
		Iteration duration
		No. of issues at start time
		No. of issues added
		No. of issues removed
		No. of to-do issues
		No. of in-progress issues
		No. of done issues
		No. of Scrum masters
		No. of Scrum team members
		Planned stories completion ratio
		Unplanned new points completed ratio
		Average speed to resolve issues
		Persons-month (PM) in the sprint
		Size of the code changes for an issue
		No. of Open issues to be done in remaining time of the sprint
		Feature throughput
		Release frequency
		Productivity rate
		Beta-testing time
		Acceptance testing time
	Issues' Estimation Accuracy	Accuracy of planning effort of issues
Blocking	Issues' Definition Completeness	No. of issues incomplete in the product backlog
	Delayed Issues	No. of blocked tasks
		No. of blocking tasks
		No. of open issues whose created time is older than a month

The QM shown in Table 1 was presented to the four UC providers during the first GQM workshop. However, it has to be noted that it underwent evolution until M15 for the PoC. This is the reason because there might be a slight mismatch between current Q-Rapids QM and the Q-Rapids QM that was used in this activity.

2.5 GQM workshops implementation

The agenda of the GQM workshops for each UC considered five steps to carry out the tasks described in Figure 1:

- Step 1: characterizing the environment/scope in which the metrics for analyzing Q-Rapids software development process will be used in the UC.
- Step 2: reviewing measurement goals for analyzing Q-Rapids software development process.
- Step 3: analyzing suitability of existing product and process factors in Q-Rapids QM, and eliciting new factors/metrics when needed.

Copyright © Q-Rapids consortium – All rights reserved

- Step 4: setting a baseline for factors/metrics identified in Step 3.
- Step 5: defining a measurement plan.

In practice, besides off-line work exchanged via email, three GQM sessions were conducted with each UC provider (Bittium, ITTI, Nokia, and Softeam) in order to elicit the metrics that make sense in each company case to meet the goals of T2.2, to create a measurement plan, and to set-up a baseline for each metric. An overview of the GQM workshops carried out in each UC is presented in Appendix B.

2.6 Cross-case analysis: aggregation of metrics

Figure 4 illustrates the overall flow of activities to aggregate the metrics identified in the GQM workshops:

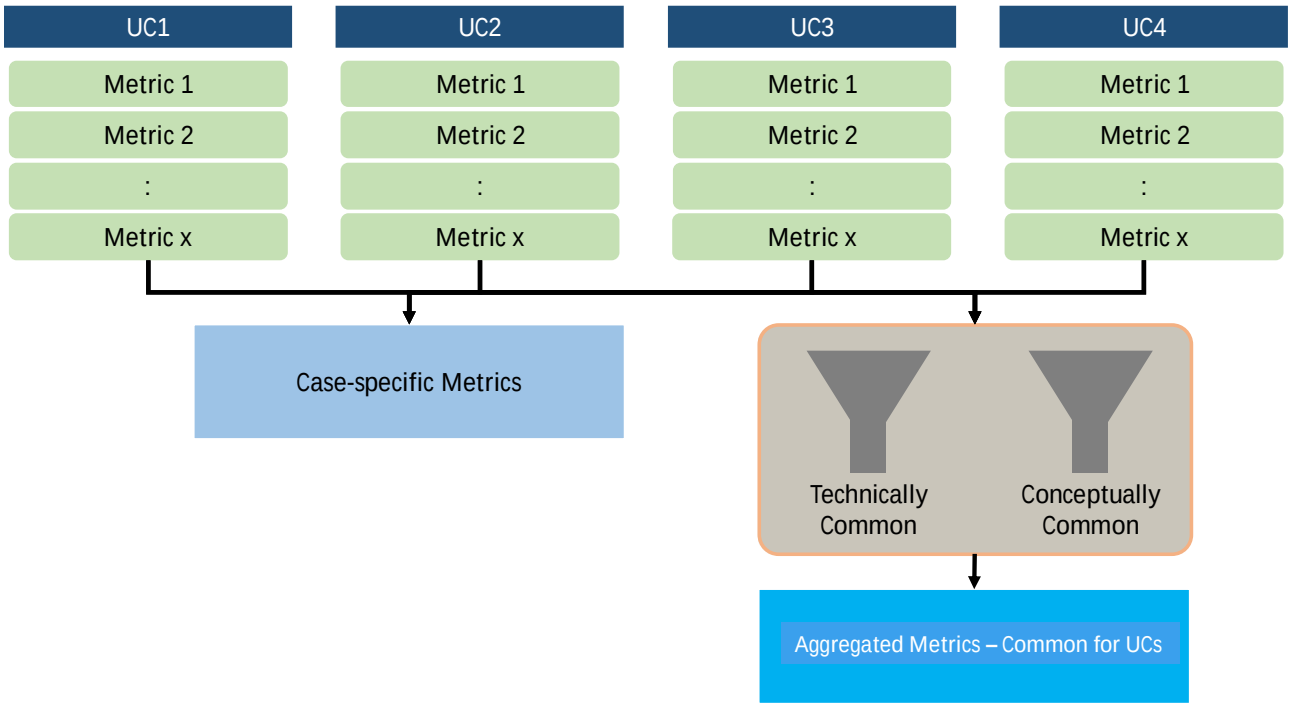

Figure 4 Flow for aggregating Metrics

Following steps were taken from the standpoint of aggregating baseline process metrics:

1. Individual metrics were collected from every UC provider, without influencing their choice by what other UC providers have included, or what researchers at the University of Oulu (UOULU) would like these companies to measure. However, these companies were provided with Quality Factors/Metrics already existing in Q-Rapids QM to analyze if they could be used as key quality and performance indicators in the context of T2.2.
2. The accumulated list of metrics was analyzed by researchers at the UOULU for commonality. The comparisons were twofold, viz. based on technical commonality and conceptual commonality. The first distinction relates to two or more UC providers interested in measuring identical metrics. Conceptual commonality refers to metrics that may differ in how they are measured, but they bear conceptual similarities to metrics preferred in other cases. For instance, a company may be interested in measuring "Error leakage", which measures the share of errors that was no accounted for in previous release. On the other hand, another company may be interested in measuring "No. of errors identified after external delivery (major release)", which are basically errors that are

Copyright © Q-Rapids consortium – All rights reserved

reported after a product is deployed commercially. In both these above cases, the common goal is to account for errors that escaped the development team up until the major release.

3. The outcome of the analytical framework highlighted in the 2nd step above is the formulation of set of aggregated metrics that are common among UC providers (two or more) and metrics that are UC-specific.

Copyright © Q-Rapids consortium – All rights reserved

3. Measurement Plan

This section presents the measurement plan outcome of T2.2. It outlines the activities that will be performed to measure Q-Rapids process performance and quality against baseline values provided by the UC providers, and the stages of storage, analysis, and visualization that data that will be put through.

3.1 Measurement goals

Two measurements goals were defined based on the description of T2.2 as presented in Table 2. These measurement goals guided the elicitation of metrics during the GQM workshops.

Table 2. Measurement Goals

Measurement Goal	Description
Measurement goal 1	Analyzing the Q-Rapids software development process for the purpose of monitoring with respect to process performance from the viewpoint of the Q-Rapids process users in the context of the Use Case
Measurement goal 2	Analyzing the Q-Rapids software development process for the purpose of monitoring with respect to process quality from the viewpoint of the Q-Rapids process users in the context of the Use Case

3.2 Measurement context

During the environment characterization in which the metrics will be used in each UC, we learned that, in reality, environments are dynamics. Therefore, although it is not expected that measurement environments will dramatically change along the time-life of the Q-Rapids project, it is unrealistic to try to keep them as a constant. It is expected that the environments in which the metrics will be used will somehow evolve during the course of Q-Rapids project (scope in which we will compare the software development process before and after using Q-Rapids solutions). For example, in the case of Bittium, Q-Rapids solutions will be initially piloted in a small team developing an internal product, but it is expected that the environment will be extended as Q-Rapids solutions mature and the company starts getting benefits out of using these solutions. Nokia plans to test Q-Rapid solution's effectiveness against a large-scale commercial product, involving globally distributed development team (squad group). Due to the nature of the project, the measurement context will evolve, which will further allow us to fine-tune Q-Rapid solution to adapt to the varying QR measurement demands. In case of ITTI, no single project will remain common throughout the Q-Rapid project. The project duration at the company has an average of about 2.5 months. This means, the company will be deploying Q-Rapid solutions under different project contexts throughout the Q-Rapid project timeline. The team constitution may change marginally, as the core team is retained across different projects. This presents a great opportunity to test the solution's adaptability, amid the changing measurement environment. Softeam will be deploying the Q-Rapid solution for an existing product line of a model-driven tool suite, released biannually. The modeling tool itself is constituted by multiple products and extensions required to be operated on a varied set of platforms. The diverse requirements based on platforms and the type of products within the suite will demand for a Q-Rapids solution that adapts to the distinct QRs, influencing process performance or quality.

However, this does not mean that the environment cannot be characterized but, contrary, that we need to put special attention to the environment characterization phase. As presented in Figure 5, we need to carefully control the environment in which the metrics are being measured in order to properly analyze the results.

Copyright © Q-Rapids consortium – All rights reserved

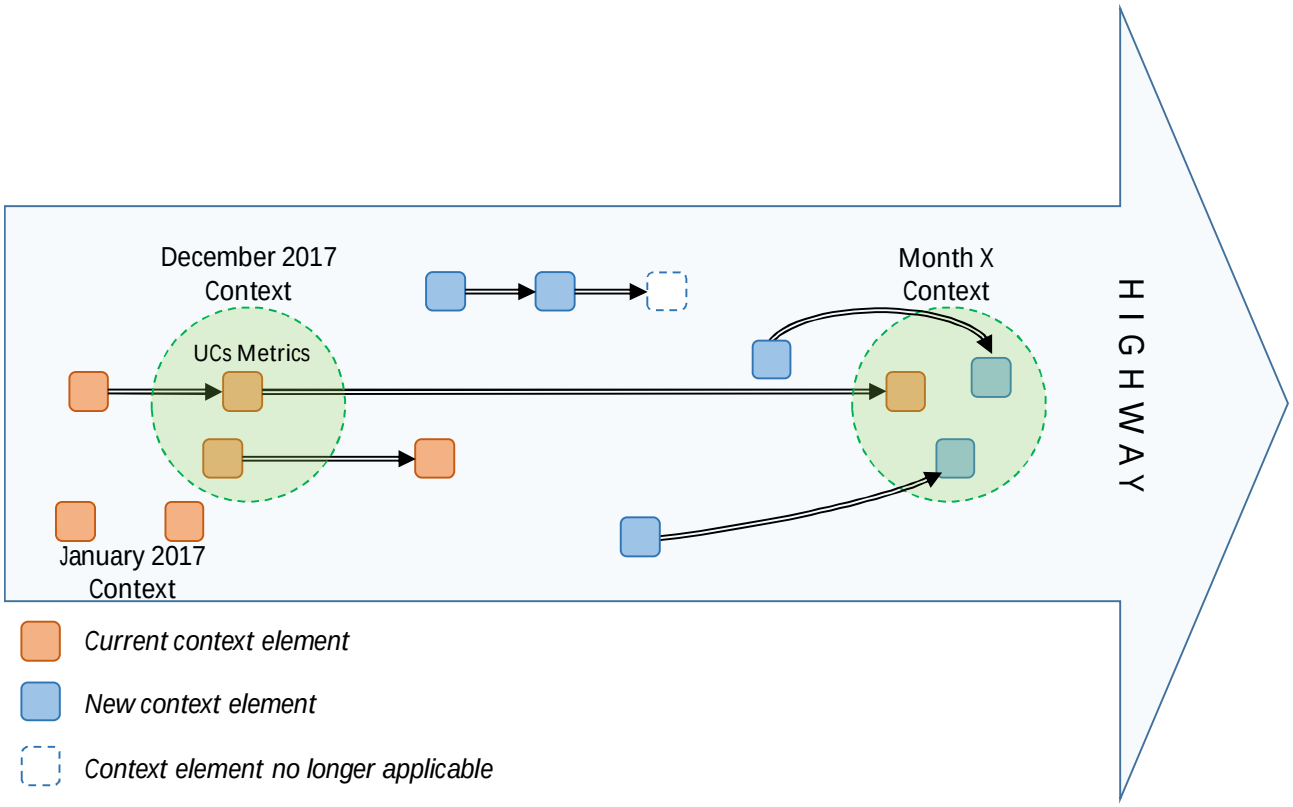

Figure 5. Evolving context of a Project

"Highway" in the above depiction of context evolution represents the overall dynamic environment at a company that influences its software development processes and the measurement environment. The environment, for multitude of foreseeable and unforeseeable reasons, could change. The aim should be to adapt to those changes while undertaking process measurements, and to define a context for measurement that is maintained within manageable limits of context changes. In addition, it is expected that different metrics will be needed once the scope is extended. Therefore, the set of metrics used to monitor Q-Rapids software development process needs to be monitored as well during the project to see whether new metrics need to be added or some metrics need to be dropped.

Therefore, context will be carefully described at each measurement point to properly analyse the results. Contexts for the baseline provided at M15 have been carefully described for each UC in Section 4.1.

3.3 Baseline Process Metrics – "Metrics to Improve"

This section presents the list of metrics that were elicited during the GQM workshops and will be used as a base in the context of T2.2. Baseline process metrics are distinguished across two categories, viz. metrics that are common among UCs (at least two UCs) and metrics that are UC-specific. A third set of metrics is also provided, which the UC providers have shown interest in, but do not have the means at present to measure them, and would consider measuring them in future. Some of these metrics are hypothetical in nature, and may undergo evolution or be dropped completely post intensive analysis by the respective companies and the researchers at UOULU.

In addition, it is also worth noting that we did not classify the identified metrics on quality factors or strategic indicators at this point but only classified them under "Category", which is a temporary representation of the former. Categories identified during GQM workshops need further analysis before they can be integrated in the Q-Rapids QM (following the structure of Strategic Indicator, Quality Factor and Metric). This is primarily

Copyright © Q-Rapids consortium – All rights reserved

true for factors and strategic indicators that were not part of the existing Q-Rapids QM, but were identified in the workshops (new metrics elicited during GQM workshops). Baseline process metrics will be integrated into the Q-Rapids QM for new versions of the QM, and formulated into appropriate quality factors and/or strategic indicators.

3.3.1 Baseline Process Metrics – Common for the UCs

Table 3 lists all the baseline process metrics that are common among UCs. The commonality could be among two UCs or more. Among the baseline process metrics that were presented to the UC providers at the first Q-Rapids workshops (Table 1), not all of them were eventually found relevant and measured for T2.2. Therefore, the final set of metrics reported in D2.2 is a combination of some metrics from the Q-Rapids QM, mainly a subset considered for PoC evaluation (see D1.2), and new metrics that surfaced in the course of ensuing GQM workshops.

In Table 3, unless specified, "issues" is how a feature/story point is represented in Jira.

Table 3 Baseline Process Metrics common among UCs (at least 2)

Category	Metric	Bittium	ITTI	Softeam	Nokia
Testing Performance	Unit Test Duration	P			P
	Time spent between when an error is identified and when it is corrected	P		P	
	Error Leakage	P		P	
Issues' Velocity	Time when an issue/ticket/feature is started	P		P	P
	Throughput time for an issue/ticket/feature			P	P
	Average speed to resolve issues	P		P	
	No. of issues/tickets at start of the sprint	P		P	
	No. of issues/tickets added during the sprint	P	P	P	
	No. of issues removed during the sprint	P		P	
	No. of done issues at the end of the sprint	P		P	
	No. of in-progress issues/tickets at the end of the sprint	P		P	
	No. of unresolved issues/tickets with due date in this sprint		P	P	
Code Quality	Complexity	P	P	P	P
	Files under the threshold of complexity (%)	P		P	P
	Files exceeding the comments percentage threshold (%)	P		P	P
	Files under the duplicated lines percentage threshold (%)	P		P	P
	No. of Issues (violations)	P		P	P
	Technical Debt	P		P	P
	Lines of code	P		P	P
Testing Status	No. of Project (old jobs)	P			P

Copyright © Q-Rapids consortium – All rights reserved

	No. of Builds	P			P
	Testing stability of a project	P			P
	No. of Unit tests	P			P
	No. of Unit test errors	P			P
	No. of Unit test failures	P			P
	Unit test success density (%)	P			P
Blocking Code	% files without critical/blocker quality rules			P	P

Overall, UC providers are interested in measuring "Testing performance" and "Issues' velocity". In addition, both Bittium and Nokia identified "Code quality" and "Testing status" as relevant for analyzing Q-Rapids process quality and performance.

3.3.2 Baseline Process Metrics – UC-specific metrics

In addition to common metrics, there were also metrics that UC providers deemed relevant from the viewpoint of influencing software development process performance and quality in their specific case. This is natural as, although the software development processes of all UC providers based on agile software development, each UC provider has its own particularities. The following tables list UC-specific metrics for each UC provider (Bittium, ITTI, Softeam, and Nokia):

Table 4 Baseline Process metrics specific to Bittium.

Category	Metric
Testing Performance	% Tests under the threshold of duration
	Lead time to the first response from a developer for a commit
	Commit Size
	No. of iterations in the code review phase for a commit
	Average number of iterations in the code review phase
	Total number of identified errors in previous release
	% of errors identified in daily build (developer)
	% of errors identified during the sprint
	Average time to fix an error
Issues' Velocity	Team throughput velocity
	No. of updated issues in a given period
	No. of open issues whose created time is older than a threshold (e.g. a month)
Development Speed	Daily build pass through time
	Commit review speed
	Automation speed
Testing Status	Tests passed (%)

Table 5 Baseline Process Metrics specific to ITTI

Category	Metric
Testing Performance	No. of tickets in the "Ready" column
	No. of tickets in the "Testing" column
	No. of already tested tickets
	No. of tickets of type "bug"

Copyright © Q-Rapids consortium – All rights reserved

	No. of closed "Closed" tickets
	No. of tickets that failed tests
	No. of tickets that are pending tests
Issues' Velocity	No. of issues in "Sprint Backlog"
	No. of "Ready" issues
	No. of "Work in Progress" issues
	No. of issues that are delayed
	Total number of issues
	No. of issues that are Closed
	No. of issues that have been Tested
	No. of issues that have been Opened
Code Quality/Factor 1	Comment Ratio
	Duplication Density
Code Quality/Factor 2	Code Reliability
	Code Security
	Code Maintainability
Testing Status	Non-Bug Density
	Test Coverage
	No. of Tests Done

Table 6 Baseline Process Metrics specific to Softeam

Category	Metric
Testing Performance	Total number of bugs for a given release
	No. of bugs identified before delivery date for a given release
	% of errors identified during validation for a given release
	No. of issues that are reopened by testers
	# feedbacks provided by testers to developers on the same issue (communication cycles between developers and testers for the same issue)
	% fast tests builds
On-time Delivery	No. of releases in a given period (per year)
	No. of releases delivered on time in a certain period (e.g. year)
	% Releases delivered on-time in a given period (e.g. year)
	No. of features specification during the development cycle
	No. of features specification that are delivered on time during the development cycle (6 months cycle)
	% features specification delivered on-time during the development cycle
	No. of features in a given period
	No. of features that are delivered on time in a certain period
	% features delivered on-time during the development cycle
	No. of commit on core component in a given period (1 month) before the planned release.
	No. of commits that are not related to an identified issue in a given period (1 month) before the planned release.

Copyright © Q-Rapids consortium – All rights reserved

Product Quality	No. of feedbacks from end-users related to issues in a given period (per month)
	No. of times that end-user calls to support team in a given period (per month)
	No. of post in the open source forum in a given period

Similar to Bittium, Softeam is able to either measure above metrics using manual methods, or they have system in place to derive an expert estimate from. It is their educated assessment that these metrics can be collected in future if appropriate connectors are deployed for the purpose.

Table 7 Baseline Process Metrics specific to Nokia

Category	Metric
Testing Performance	Total number of test cases executed per issue/feature
	Actual feedback time from CI to developers (from the beginning of test run until its end)
Testing Status	Condition coverage
	Line coverage
Quality Issues Specification	% issues completely specified

"Testing Status" and "Quality Issues Specification" are metrics defined in Q-Rapids QM. Meaning, these could have been collected by all the UC providers. However, only Nokia found these relevant from their development process performance and quality point of view, and were able to measure these using the connectors implemented for the PoC.

3.3.3 Baseline Process Metrics to be considered

Besides the metrics presented in Sections 3.3.1 and 3.3.2, other metrics emerged as relevant for measuring Q-Rapids process performance and quality during the GQM workshops. However, these metrics did not appear to be the current priority for the UCs. Therefore, they would be considered for measurement in future, if needed. Next tables present these metrics per UC.

Table 8 Process metrics to be considered for Bittium

Category	Metric
Testing Performance	% of errors identified after the sprint review
	% of errors identified after internal delivery (minor release)
Development Speed	Time between a developer makes a commit and s/he receives feedback from the CI system (commit pass through time)
	Time when a commit is put for review
	Time when the commit is reviewed
	Time when the commit goes for automatic build and testing
	Time when the commit is fully tested
	Total commit gating time

Metrics in above Table 8 have been analysed and shortlisted for future evaluation because, at present, the necessary system to be able to measure these metrics is not in place (like Gerrit), nor the right connector is

Copyright © Q-Rapids consortium – All rights reserved

available. Based on Bittium's current estimate on measuring these metrics, they foresee heavy reliance on connectors for Gerrit.

Table 9 Process metrics to be considered for ITTI

Category	Metric
Issues' Estimation Accuracy	Estimated effort of an issue/story point
	Real invested effort of an issue/story point
	% Estimations under the threshold of accuracy (density of estimations exceeding the threshold of accuracy)
	Estimated effort of a sprint vs. real invested effort of a sprint (sum of all ticket efforts)
Delayed Issues	No. of open issues whose created time is older than a threshold (e.g. a month)

Table 9 are based on ITTI's current assessment of what metrics will be of relevance to the Company's development process performance and quality. There is a certainty in regard to measuring these metrics in future, but reformulation of measurement approach or modifications in the metrics themselves are very likely.

Table 10 Process metrics to be considered for Softeam

Category	Metric
Issues Estimation Accuracy	Estimated effort of an issue
	Real invested effort of an issue
	Inaccurate estimations
	% Estimations under the threshold of accuracy (density of estimations exceeding the threshold of accuracy)
	No. of issue estimations in a given period (e.g. sprint)
On-time delivery	Release due date
	Release delivery date
	features specification due date
	features specification done date
	feature due date
	feature delivery date

High-level metrics, such as' % releases delivered on-time in a given period (e.g. year)' are currently collected using informal means (excel/word/meeting, etc.). Low-level metrics such as 'release due date' and 'release delivery date' are not being systematically collected for the moment. Softeam plans to experiment with formal tools and methods going forward.

Table 11 Process metrics to be considered for Nokia

Category	Metric
Testing Performance	Average throughput time in a comment from reviewers
	No. of iterations in the code review phase for a commit
	Time when an error is identified
	Time when an error is corrected

Copyright © Q-Rapids consortium – All rights reserved

	Time spent between when an error is identified and when it is corrected
Issues' Velocity	No. of issues/tickets at start of the sprint
	No. of issues/tickets added during the sprint
	No. of issues removed during the sprint
	No. of done issues at the end of the sprint
	No. of in-progress issues/tickets at the end of the sprint
	No. of unresolved issues/tickets with due date in this sprint
	Time when the issue is started (e.g. in JIRA)
	Time when the issue is closed (e.g. in JIRA)
	No. of hours that an issue is under development
	Average speed to implement an issue
	Issue size (size of the code changes for an issue)
Issues' Estimation Accuracy	Estimated effort of an issue/story point
	Real invested effort of an issue/story point
	% of correct estimations
	% of incorrect estimations > +/- 20
	Estimated fbxx.yy for an issue (fb= feature build, x= year, y= month)
	Actual fbxx.yy+1 to deliver the issue
	% of estimations correct
	% of estimations incorrect > +/- 2

Drawn from current development processes and way of managing QRs, most of the metrics listed in Table 11 above have been reformulated by Nokia. This is a testament to Nokia's interest in these metrics and the preliminary assessment of being able to collect their values.

UC providers are mostly confident about the metrics they have allocated for future consideration. Softeam has reservations about the metrics they plan to consider in future, as these metrics need further assessment and appropriate measurement approach also need to be formulated for them. Just like in the case of Softeam, some of the metrics for "future consideration" may undergo alterations depending on the changes in context.

3.4 Data collection and storage mechanisms

In general, each UC provider will make use of Q-Rapids existing connectors and will adapt them to meet the measurement requirements of the metrics they are interested in. Such an activity will require close collaboration between UC providers, WP2 and WP1. WP1's willingness to allow UC providers to customize these connectors will enable swift and hassle-free metrics value collection. UC providers will also have the option of developing a connector exclusive to their own requirements. Similarly, there are no disparate plans to store the values for the metrics collected under T2.2. Storage mechanism used for data collected for Q-Rapids QM for the PoC will be retained for the storage of other metrics as well.

3.5 Data analysis

The gathered metrics and their baseline values will serve analytical purpose on two fronts, discussed as follows:

1. Each UC provider can use the data and the concomitant insights to engineer in-house software development process improvements, leading to enhancement in QRs management in their UC. The changes in the baseline values over the entire duration of the Q-Rapids project will operate as an

Copyright © Q-Rapids consortium – All rights reserved

indicator, as these companies fine-tune how they collect QRs and implement them. Researchers from UOULU are available to assist these companies in this activity. The outcome of this activity will serve as an input for T2.3 and T2.4.

2. Researchers from UOULU will evaluate these metrics and their values, along with regular updates from the companies, to derive the overall effectiveness of Q-Rapids solution (T2.3 and T2.4). UOULU will evaluate the effectiveness of Q-Rapids solutions, and how the insights it entails could help affect process performance and quality, and if the companies are able to extract benefits from it. Furthermore, the distinct environment and variations in the same will allow the researchers to develop a dynamic Q-Rapids solution that accounts for context changes.

3.6 Measurement results visualization

All the metrics and their values will be monitored and visualized using the Q-Rapids Dashboard (in collaboration with WP3). The current Dashboard will undergo further customization to accommodate the updated set of metrics.

3.7 Measurement schedule

Table 12 depicts the measurement schedule for the Q-Rapids process metrics elicited, and the various applications of the resulting data based on the Q-Rapids project objectives and timelines outlined in the DoA.

Table 12 Measurement Schedule

Q-Rapids Process Metrics Aims and Applications	Project Months						
	M15	M16	M20	M25	M30	M33	M36
Continuous measure, evaluation, analysis and interpretation of process metrics data							
Analysis of the impacts of piloting Q-Rapids methods and tools to QRs lifecycle process in use cases							
Q-Rapids process quality and performance is validated by comparing Q-Rapids process performance and quality to baseline							

Baseline process metrics from M15, part of D2.2, will be further expanded and interpreted throughout the project lifespan. Since UC providers will be measuring these metrics on a regular basis, informing them on level of their process performance and process quality, this activity will continue until M36. From M16 to M33, impact of the transformation in software development processes induced by Q-Rapids solution will be evaluated (T2.3). Between M25 and M36, best practices, tools and methods identified in T2.3 will be integrated, and comparison of process metrics from T2.2 will be undertaken (T2.4). Empirical validation of the impact of Q-Rapids solution and baseline process comparisons will also be conducted during this period.

Copyright © Q-Rapids consortium – All rights reserved

4. Baseline (M15)

Based on the three workshop sessions with the UC Providers, a set of metrics was identified whose values could be acquired for M15 (see Sections 3.3.1 and 3.3.2). Hard data as well as expert estimates were used to create a baseline for these metrics. Wherever possible, connectors were used to collect hard data. In cases this was not feasible, baseline values were collected using manual methods by querying current system or combination of systems to collect the needed data. As a last resort, where neither approach was possible, expert estimates were used to provide these baseline values.

Taking into account the disclosure preference for the baseline data, the baseline values are available for perusal in Appendix C. This appendix is privately available only for Q-Rapids partners and Commission reviewers. Once this document is made publicly available, the appendix must be deleted from the document.

4.1 Measurement Contexts

Baseline values for the current deliverable are provided within the measurement contexts outlined in Table 13:

Table 13. UCs' measurement environments for M15

Dimension	Description
Bittium	
Project	Bittium's Q-Rapids UC - Devices
Product	Production Testing SoftWare framework (PTSW). Internal product created to verify HW (factory automation software). This product is used when verifying the HW of the Tough Mobile.
Software development team	One team composed by 6-7 members
Software development process	Scrum
Stakeholders interested in these metrics	The team in which Bittium is piloting Q-Rapids solutions (people developing and managing the solution).
ITTI	
Project	ITTI's Q-Rapids UC (different projects of 2-3 months duration).
Product	"R"[1] Project
Software development team	One scrum team (always the same core team although some people may change)
Software development process	Scrumban for visualizing tasks Scrumbut as software development process. Software development process based on Scrum but with some exceptions
Stakeholders interested in these metrics	The team in which Q-Rapids solutions will be piloted and its Product Owner.
Nokia	
Project	Nokia's Q-Rapids UC
Product	Baseband Platform
Software development team	· Squad group: MCU HWAPI 1 (15 developers, squad group leader; Finland) · Squad group: MCU HWAPI 2 (11 developers, squad group leader; Finland) · Squad group: MCU HWAPI 3 (25 developers, squad group leader; Poland)

[1] Fictitious name. Real name is not reported due to confidentiality issues.

Copyright © Q-Rapids consortium – All rights reserved

	· Squad group: MCU HWAPI 4 (22 developers, squad group leader; Poland) · Squad group: MCU HWAPI 5 (13 developers, squad group leader; Poland) · Squad group: MCU HWAPI 6 (14 developers, squad group leader; France) · Squad group: MCU HWAPI 7 (9 developers, squad group leader; USA) · Squad group: MCU HWAPI 8 (8 developers, squad group leader; USA) In addition, Nokia may engage any other relevant stakeholder in the evaluation tasks e.g. quality managers, product managers and product owners
Software development process	Scrum
Stakeholders interested in these metrics	Product Owners and SW developers, Quality Managers, Product Managers.
Softeam	
Project	Modelio Case: Monitoring the development of a biannual release of Modelio Modeling tool.
Product	Modelio Modeling Tool and its extensions
Software development team	A team composed by 9 people (Modelio development team).
Software development process	Ad-hoc process based on Agile principles. Model oriented development
Stakeholders interested in these metrics	The team in which Softeam is piloting Q-Rapids solutions (people developing and managing Modelio) in order to show how well they are doing.

4.2 Baseline values (M15)

Baseline values are provided in Appendix C. UC providers, with their distinct measurement context, development processes, and connector requirements, collected baseline values under certain conditions. These conditions have been documented in Table 14. The parameters convey the following meaning from measurement context perspective.

1. Baseline value type – Indicates if the baseline value was collected at a particular instance during software development (One-time), or has been aggregated over a period of time (Average). Each UC provider selected the type that they considered more appropriate to provide a reference value to be used as baseline. In case of 'on-time' baseline value type, UC provider picked a particular instance during the course of the project development that was representative of the entire project.
2. Unit of timeframe for baseline value – The unit of timespan that was considered for collecting the baseline value. This could either be part of an entire project *(# Sprint(s)*), or the complete project itself (*Entire project*).
3. Duration of the timeframe – Total time period covered by the unit of timeframe considered in case of each UC provider. The duration could span from weeks to months, or even years.
4. Stage/Phase of the project during the timeframe – Distinguishing the exact instance in a project from where the baseline value was collected. This could mean a rudimentary division of the entire project (*Beginning, Middle, End*), or a custom stage/phase that's defined by a UC provider.

Copyright © Q-Rapids consortium – All rights reserved

5. Team retention for the timeframe – Depicting if an officially allocated team for the project was retained throughout the duration (*Retained*), or if there were changes in the team (*Changes*).
6. Other remarks – Other relevant ancillary remarks.

Table 14 Baseline value collection conditions

Baseline value conditions	Bittium	ITTI	Softeam	Nokia
Baseline value type	Average	One-time	Average	Average
Unit of timeframe for baseline value	1 Sprint	Entire Project	3 Sprints	169 sprints
Duration of the timeframe	2 weeks	8 months	6 months	22.5 months
Stage/Phase of the project during the timeframe	Beginning	Middle	End	Ongoing (Continuous Development)
Team retention for the timeframe	Retained	Retained	Retained	Changes
Other remarks	Baseline from an internal project (PTSW)	Baseline from project repositories using internal plugins	Baseline value from the base modeling tool and its extensions	Baseline values from features developed by eight "Squad Groups" functioning as a single team

The additional information included in Table 14 will help monitor and compensate for the changes in the measurement context in future. This will help draw legitimate comparisons between the baseline values and the values collected in the future, where the changes in the context have been taken into consideration, allowing for a more meaningful interpretation of the outcome.

Excluding ITTI, in which baseline values were created based on the middle of one of the ITTI's projects ('R' project), baseline values for the UC providers were computed based on averages in M15, taking into account their specific unit of timeframe and duration. In case of Bittium, baseline values were collected in the two weeks leading up to M15. "Beginning" phase of the timeframe means that Bittium considered the sprint at the beginning of the project. Similarly, it was 6 months leading up to M15 when Softeam collected their baseline values, which corresponded to the end phase of the project. In case of Nokia, the timeframe considered for collecting the baseline values was 22.5 months. Nokia engages in Continuous Development (CD), so a project cannot be distinctly classified across traditional phases of "beginning", "middle", or "end".

Copyright © Q-Rapids consortium – All rights reserved

Conclusion

UC providers have been active in collaborating with UOULU to derive the metrics that could monitor the quality and performance of their Q-Rapids software development process. The resulting metrics, after careful analysis for commonality, seem like a good stepping stone towards measuring Q-Rapids process performance and quality, supporting T2.3 and T2.4. The maturing Q-Rapid solution, and the gradually deepening project understanding and the process improvement potential it holds would enable all the involved stakeholders to elicit further metrics, if needed. For example, while some metrics will be very useful to assess Q-Rapids process performance and quality and guide improvement actions, others might not provide the expected insights. Therefore, Deliverable 2.2 will be updated and new versions will be released at M24 and M33 to reflect possible changes in process metrics.

Copyright © Q-Rapids consortium – All rights reserved

References

[1] V. R. B. G. Caldiera and D. Rombach, "The goal question metric approach," *Encyclopedia of software engineering,* vol. 2, pp. 528-532, 1994.

[2] V. R. C. G. &. R. H. D. Basili, "Experience factory.," in *Encyclopedia of software engineering.*, 1994.

Copyright © Q-Rapids consortium – All rights reserved

Appendix A – Detailed Q-Rapids QM

The following table presents the metrics from the Q-Rapids QM presented to the UC providers at the first GQM workshop. These metrics are classified as per the Factors they fall under in the QM.

Table 15 Q-Rapids QM from the first GQM Workshop

Metric	Metric Description	Unit
Code Quality		
Complexity	It is the complexity calculated based on the percentage of files that exceed a defined average complexity per function	Number
Files under the threshold of complexity (%)	Density of files not exceeding the threshold of complexity	%
Cognitive Complexity	How hard it is to understand the code's control flow.	Number
Files under the threshold of cognitive complexity (%)	Density of files not exceeding the threshold of cognitive complexity	%
Files exceeding the comments percentage threshold (%)	Percentage of files with a recommended density of comment lines	%
Files under the duplicated lines percentage threshold (%)	Percentage of files with a recommended density of duplication	%
Issues (violations)	No. of issues (violations) based on code quality rules.	Number
Technical Debt	Effort to fix all maintainability issues.	Time (hours, days, weeks, etc.)
Lines of code	No. of physical lines that contain at least one character which is neither a whitespace nor a tabulation nor part of a comment.	Number
Testing Status		
Condition coverage	Has each Boolean expression been evaluated both to true and false?	%
Unit tests violating the condition coverage threshold (%)	Share of unit tests among all the unit tests that do not satisfy the predefined condition coverage threshold	%
Line coverage	Has this line of code been executed during the execution of the unit tests?	%
Unit tests violating the line coverage threshold (%)	Share of unit tests among all the unit tests that do not satisfy the predefined line coverage threshold	%
No. of Projects (old jobs)	No. of projects (user-configured descriptions of work which Jenkins should perform)	Number
Projects passed (%)	Density of builds failed	%
No. of Builds	No. of builds (results of a single execution of a project)	Number
Testing stability of a project	Tendency in the last 5 builds of a project	Number
No. of Unit tests	No. of unit tests.	Number

Copyright © Q-Rapids consortium – All rights reserved

No. of Unit test errors	No. of unit tests that have failed.	Number
No. of Unit test failures	No. of unit tests that have failed with an unexpected exception.	Number
Unit test success density (%)	Test success density	%
Stability		
No. of bugs and errors in the current sprint	No. of issues with status "open, in progress, or reopened" of type "bug or error" with priority higher or equal to "medium"	Number
Customer satisfaction	No. of complaints after product release (e.g., from reviews, hot line, etc.)	Number
Testing Performance		
Unit tests duration	Time required to execute all the unit tests.	Time (hours, days, weeks, etc.)
Tests under the threshold of duration (%)	Density of running test not exceeding the threshold of duration (e.g., 5 min)	%
Issues' Velocity		
Effort needed for unresolved issues with due date in this sprint / Remaining effort in the sprint	Effort required to implement the pending issues planned for the current sprint	Time (hours, days, weeks, etc.)
No. of unresolved issues with due date in this sprint	Issues that remain untouched after the sprint has ended, indicating at improper allocation/utilization of resources.	Number
Iteration duration	Measure of time spent on an issue/ticket	Time (hours, days, weeks, etc.)
No. of issues at start time	Measure of the number of issues at the beginning of a sprint	Number
No. of issues added	Measure of the number of issues added while a sprint is in progress	Number
No. of issues removed	Measure of the number of issues that were removed while sprint is in progress	Number
No. of to-do issues	Measure of the number of issues under "To do" column	Number
No. of in-progress issues	Measure of the number of issues under "In Progress" column	Number
No. of done issues	Measure of the number of issues that are "Done"	Number
No. of Scrum masters	Total number of scrum masters for a sprint	Number
No. of Scrum team members	Total number of scrum team members for a sprint	Number
Planned stories completion ratio	Ratio of stories completed during a sprint to the total stories planned for that sprint	Ratio
Unplanned new points completed ratio	Ratio of unplanned story points completed in a sprint to total story points not considered for that sprint	Ratio

Copyright © Q-Rapids consortium – All rights reserved

Average speed to resolve issues	Average of end time of each issues - start time of each issue	Time (hours, days, weeks, etc.)
Persons-month (PM) in the sprint	Measure of total effort in persons-month for a sprint	Person-month
Size of the code changes for an issue	Total number of code lines changed for an issue	Number
No. of open issues to be done in remaining time of the sprint	Total number of "To do" issues to be completed in the remaining time of a sprint	Number
Feature throughput	Percentage of features that meet time to market target with the desired levels of quality	%
Release frequency	No. of features released per time unit (e.g., year)	Number
Productivity rate	Time used for development and test of new features / time used for maintenance or defect removal	Time (hours, days, weeks, etc.)
Beta-testing time	Time from start to end dedicated to beta-testing (feature-specific)	Time (hours, days, weeks, etc.)
Acceptance testing time	Time from start to end dedicated to acceptance test (feature-specific)	Time (hours, days, weeks, etc.)
Issues' Estimation Accuracy		
Accuracy of planning effort of issues	Estimated effort (hours or story points) of an issue vs. real invested effort (hours or story points) of an issue	Time (hours, days, weeks, etc.)
Issues' Definition Completeness		
No. of issues incomplete in the product backlog	No. of issues in the backlog with specification definition incomplete	Number
Delayed Issues		
No. of blocked tasks	Total number of tasks that are blocked by other tasks	Number
No. of blocking tasks	Total number of tasks that are blocking other tasks	Number
No. of open issues whose created time is older than a month	Total number of issues that were created more than a month ago, but still pending implementation	Number

Copyright © Q-Rapids consortium – All rights reserved

Appendix B: GQM Workshops Overview

The following table presents the details of the GQM workshop sessions conducted with each UC provider, along with the outcomes of each session:

Table 16 GQM Workshop sessions conducted with each UC provider

Session	Participants	Outcomes
	Bittium	
Session 1 (3.5 hours)	Bittium: Quality and Process Lead, Tooling/Generic Tech Lead, Embedded Software Development Lead, and Requirement & Process Lead UOULU: 2 researchers	- List of data sources that could be used in Bittium - Scope partially defined (step 1) - Measurement goals revised and accepted (step 2) - Selection of existing factors form the Q-Rapids QM (step 3) - Elicitation of new factors/metrics (step 3)
Session 2 (2.5 hours)	Bittium: Quality and Process Lead, Tooling/Generic Tech Lead, Embedded Software Development Lead, and Requirement & Process Lead UOULU: 2 researchers	Discussion on the data filled in by Bittium on individual factors and metrics that Bittium worked upon as part of their homework from Session 1.
Session 3 (2.5 hours)	Bittium: Quality and Process Lead, Tooling/Generic Tech Lead, Embedded Software Development Lead, and Requirement & Process Lead UOULU: 2 researchers	- Discussion on baseline values collected for the metrics finalized by Bittium thus far - Based on the availability of data under current operational circumstances, metrics for the first deliverable were revised - Measurement plan as well as data confidentiality concerns were addressed
	ITTI	
Session 1 (3 hours)	ITTI: Product Owner, Project Manager, and System Designer UOULU: 2 researchers	- Scope partially defined - Measurement goals revised and accepted (step 2) - Description of ITTI software development process - Selection of existing factors form the Q-Rapids QM (step 3) - Elicitation of new factors/metrics
Session 2 (1.5 hours)	ITTI: Product Owner and Project Manager UOULU: 2 researchers	- Scope fully defined - Metrics revised and data sources established - Target values for metrics provided, which is what ITTI eventually would like the metrics to reflect
Session 3 (1.5 hours)	ITTI: Project Manager UOULU: 2 researchers	- Scope refined - Metrics revised and finalized for first deliverable - Context around baseline values discussed and established
	Nokia	
Session 1 (3.5 hours)	Nokia: Quality Manager, Project Manager, SW Dev. Specialist, SW Dev. Manager, SW Dev. Specialist, SW Dev. Specialist, Quality Manager, and SW Dev. Manager UOULU: 2 researchers	- Scope partially defined - Measurement goals revised and accepted (step 2) - Selection of existing factors form the Q-Rapids QM (step 3) - Elicitation of new factors/metrics (step 3) - Prioritized factors that would feature in the deliverables - Nokia's preliminary nod on getting baseline values for the factors mutually agreed upon
Session 2 (2 hours)	Nokia: Quality Manager, Project Manager, SW Dev. Specialist, SW Dev. Manager, and SW Dev. Specialist UOULU: 2 researchers	- Scope further refined - Measurement goals further revised and accepted (step 2) - Concerns regarding data confidentiality discussed
Session 3 (1 hour)	Nokia: Quality Manager, Project Manger, SW Dev. Specialist, and SW Dev. Manager UOULU: 2 researchers	- Scope established - Metrics revised - Baseline values reviewed - Measurement Plan discussed and agreed upon
	Softeam	
Session 1 (3 hours)	Softeam: Architect/Developer, Project Manager, R&D Manager, DEO/Product Owner	- Scope defined - Measurement goals revised and accepted (step 2) - Selection of existing factors form the Q-Rapids QM (step 3)

Copyright © Q-Rapids consortium – All rights reserved

	UOULU: 2 researchers	- Elicitation of new factors/metrics
Session 2 (1 hour)	Softeam: Architect/Developer and R&D Manager UOULU: 2 researchers	- Scope discussed - Measurement goals reviewed and accepted - Metrics discussed
Session 3 (0.5 hour)	Softeam: Architect/Developer UOULU: 2 researchers	- Scope, metrics, and baseline values reviewed discussed - Decision on using a prioritized list of metrics for the deliverable

Copyright © Q-Rapids consortium – All rights reserved

Appendix C: Baseline (M15)

Contents removed from the public version for confidentiality purposes

Copyright © Q-Rapids consortium – All rights reserved