Definition of the On-Time Delivery Indicator in Rapid Software Development

Martí Manzano, Cristina Gómez, Claudia Ayala
Universitat Politècnica de Catalunya (UPC)
Barcelona, Spain
{mmanzano,cristina,cayala}@essi.upc.edu

Silverio Martínez-Fernández
Fraunhofer IESE
Kaiserslautern, Germany
Silverio.Martinez@iese.fraunhofer.de

Prabhat Ram, Pilar Rodríguez
University of Oulu
Oulu, Finland
{pilar.rodriguez,prabhat.ram}@oulu.fi

Marc Oriol
Universitat Politècnica de Catalunya (UPC)
Barcelona, Spain
moriol@essi.upc.edu

***Abstract*— Rapid software development (RSD) is an approach for developing software in rapid iterations. One of the critical success factors of an RSD project is to deliver the product releases on time and with the planned features. In this paper, we elaborate an exploratory definition of the On-Time Delivery strategic indicator in RSD based on the literature and interviews with four companies. This indicator supports decision-makers to detect development problems in order to avoid delays and to estimate the additional time needed when requirements, and specifically quality requirements, are considered.**

***Index Terms*— Rapid software development, On-Time Delivery indicator, Decision-making, software analytics.**

I. Introduction

Rapid Software Development (RSD) is the organizational capability to develop, release, and learn from software in rapid cycles [1]. One of the critical success dimensions of RSD projects is delivering their releases on time [2].

During RSD, a huge amount of project and development data (e.g., number of to-do issues, average speed to resolve issues, acceptance testing time) is available in several data sources (e.g., JIRA, Git, Redmine). These data may be processed and analyzed with the purpose of turning into a meaningful and relevant strategic indicator for giving decision-makers the view of how software development is going. One of the relevant information that may be assessed is the On-Time Delivery strategic indicator. We define this indicator as the capability of fulfilling the issues (feature, improvement, project task, software bug or a custom issue) planned for a specific release. We adopted the definition of issues[1] used by JIRA. The indicator provides useful information to decision-makers, as it can be used to estimate whether a team can complete a product release on time, to calculate the additional time needed when new requirements, and specifically quality requirements, are considered or to discover untracked tasks, or development problems that may lead to delays in the software delivery.

The main goal of this paper is to present an exploratory definition of the On-Time Delivery strategic indicator in RSD projects, and to identify different factors affecting the indicator (e.g. issues' due date compliance). We based this definition on the literature and feedback from four industrial cases provided by the partners of the Q-Rapids[2] European H2020 project.

The rest of the paper is structured as follows. Section 2 briefly presents the Q-Rapids general approach in which the indicator was defined. Section 3 details the research approach followed. Section 4 presents the definition and the factors influencing the On-Time Delivery indicator. Section 5 sketches the related work in the area. Finally, Section 6 summarizes the conclusions and identifies future work.

II. The Q-Rapids approach

This work is carried out in the context of the Q-Rapids European project that aims to improve the management of quality requirements in RSD processes. To achieve this goal, the project promotes a highly informative dashboard to support data-driven, requirements-related strategic decision making in rapid cycles. Q-Rapids aims to increase software quality and improve the development process through (see Fig. 1 (a)):

- Gathering and analyzing data from project management tools, software repositories, quality of service and system usage. The analysis of these data permits to systematically and continuously assess software quality using a set of relevant quality-related indicators (e.g. On-Time Delivery). Concretely, *Metrics* are computed from data gathered from data sources using software data collectors, and are elaborated into *Product/Process Factors,* based on Q-Rapids quality model [3], and ultimately aggregated into *Strategic Indicators* (see Fig. 1 (b)).

[1] confluence.atlassian.com/jira064/what-is-an-issue-720416138.html

[2] www.q-rapids.eu

DOI 10.1109/QuaRAP.2018.00006

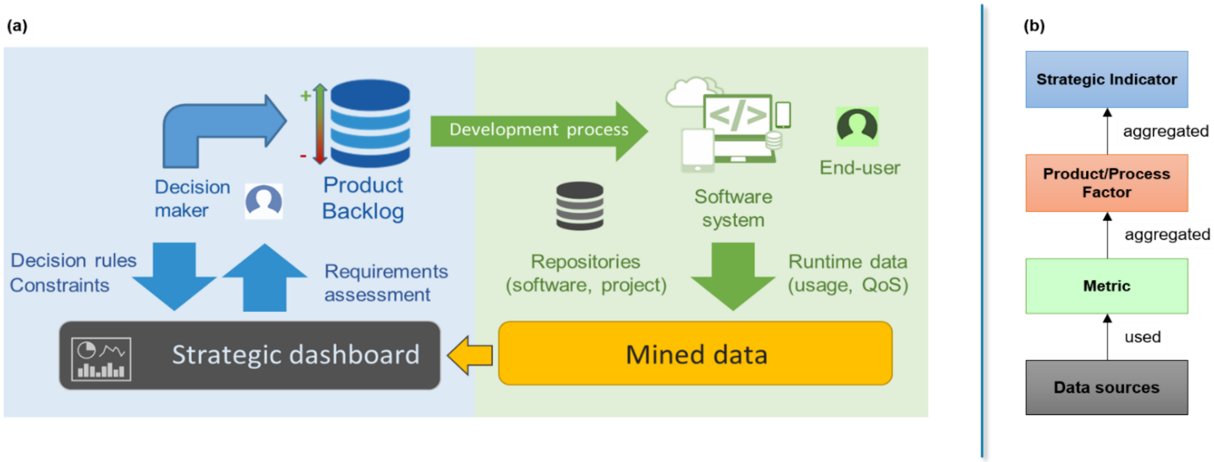

Fig. 1. (a) The Q-Rapids approach. (b) The Q-Rapids Quality Model

- Providing decision makers with a highly informative dashboard to help them make data-driven, requirements-related strategic decisions in rapid cycles. The dashboard aggregates the collected data into strategic indicators.
- Extending the agile software development process, considering the comprehensive integration of quality and functional requirements and their management, in a way that favors software quality, and that brings a significant productivity increase to the software lifecycle.

III. Research approach

In this section we elaborate the definition of the On-Time Delivery strategic indicator and identify the factors and metrics to be aggregated to assess the value of the indicator in the context of the Q-Rapids approach. We used the information provided by the four industrial partners of the Q-Rapids project, and followed the Q-Rapids quality model to define the indicator. We provide the definition of the indicator for the case of software development releases but this definition may be easily adapted and applied to the case of internal software development iterations.

The research approach followed in the Q-Rapids project is based on action-research cycles [4] composed of the following steps: 1) to identify relevant problems in the context of the industrial partners of the project; 2) to plan and perform actions to solve these problems; 3) to evaluate the effectiveness of the performed actions for overcoming the identified problems and 4) to reflect on the need of a new cycle for reaching the desired improvements. The four Q-Rapids industrial partners are from different European regions and of different sizes. They provide use cases to collect empirical data needed to ensure the industrial relevance of the addressed problems, and the effectiveness of the implemented actions, while the academic partners lead the application of the action-research cycles.

With the aim of diagnosing the most relevant and affordable problems of the four Q-Rapids industrial partners, we held several workshops, and conducted semi-structured interviews at their premises in order to identify organizational, strategic, and software product goals. In addition, we made in-situ observations, and accessed some of the industrial partners' repositories and tools to analyze the data that could be exploited. The On-time Delivery health indicator emerged from these activities as a joint need of all the Q-Rapids industrial partners.

In order to build up the complete definition of the On-Time Delivery indicator, we carried out the following activities:

1. **Initial interviews**: Specific interviews were done to identify organizational, strategic and software product goals of the industrial partners. After analyzing the results of the interviews, the need for an On-Time Delivery health indicator in RSD environments emerged.

2. **Literature review**: A literature review was performed to get a consolidated background on the indicator, and to identify the indicator breakdown: quantifiable metrics or factors that On-Time Delivery could depend on in a generic scenario. A top-down review approach was followed to extract topics that could offer information regarding the indicator. Some of the topics searched as part of the literature review include: Time-to-Market in Software Development, Software Development time and effort (and estimation of it), Effort estimation in Agile Software Development, Delivery Capability in Agile Software Development.

3. **Tentative proposal**: Deriving an extended list of methods, factors, processes, metrics, and frameworks collected from the literature review, a tentative On-Time Delivery indicator definition and its breakdown was proposed to the industrial partners.

4. **Workshops**: Two series of workshops with the industrial partners were conducted to elicit particular understandings of their use cases. These workshops were helpful to get feedback on our tentative proposal. The industrial partners were asked to add factors and metrics to the proposed breakdown, to remove them and to vote for them in order of their subjective relevance.

5. **GQM Workshops**: In addition, GQM [5] workshops were conducted with the industrial partners in an effort to elicit metrics for assessing process performance. In the process,

metrics relevant for measuring On-Time Delivery emerged in these workshops, particularly in the case of one industrial partner.

6. **Refining the proposal**: The feedback gathered from the workshops helped us to refine and build-up the On-Time Delivery indicator definition and breakdown, shown in section 4. We plan to refine the definition of the indicator during its deployment in the four use cases

IV. On-Time Delivery definition and factors

After conducting the activities described in the previous section, a consensus was reached on the definition of the On-Time Delivery strategic indicator. It is defined as the capability of fulfilling the issues planned for a specific release, meeting internal and external delivery schedules. Also, it was agreed that the values of the indicator would fall into the [0, 1] range, where 0 indicates a low capability of meeting delivery data for a specific release and 1 indicates a high capability.

In addition, as a result of the literature review and the workshops conducted with the industrial partners, we identified a set of factors that can affect the capability of delivering the committed software on-time for a specific release. These factors are: Issues' Effort Estimation Accuracy, Issues' Development Status, Issues' Due Date Compliance, and Delivery Performance. The Blocking strategic indicator, which was previously addressed in the project [6], is also considered as a factor for the estimation of the On-Time Delivery indicator. Table 1 shows the metrics for these factors, the data to gather for computing them, and their corresponding data sources. Following the Q-Rapids quality model, metrics have to be normalized using utility functions [14] to provide values from 0 to 1, and then aggregated them using weighted sums (w_{ij}) to give values to the corresponding factors. The value of the On-Time Delivery indicator is obtained aggregating the factors' values, using their corresponding weights (w_i) (see Fig. 2).

We can have a vision of the On-Time Delivery indicator through the Q-Rapids dashboard, and therefore decision-makers may perform specific actions to improve the delivery of the software. Next, we respectively explain the rationale of these factors.

First, *Issues' Effort Estimation Accuracy* indicates how reliable the current tasks' effort estimation is in terms of the differences between the past planned efforts and the actual tracked ones. This factor can have an adverse impact on

TABLE 1. Factors affecting On-Time Delivery, together with their corresponding metrics and data sources

Factor affecting On-Time Delivery	***Metric***	***Data source and information to be gathered***
Issues' Effort Estimation Accuracy	- Accuracy of planning effort of issues - Percentage of issues larger than the size threshold	Extracted from the available issues tracking systems (i.e. JIRA and Redmine). These metrics are computed as a percentage of the difference between the planned effort of past closed issues and the actual tracked effort, as well as the percentage of tasks bigger than a specified threshold
Issues' Development Status	-Ratio of the average past *velocity* and the theoretical *velocity* of the available units -Ability to resolve the remaining allocated effort -Ability to resolve the remaining unallocated effort	These metrics are computed from the product backlog (i.e. JIRA and Redmine) of the release being monitored, as aggregated information related to the development status. The release due date should be gathered to compute the remaining available time, as well as an averaged *velocity* of resolving past issues, the theoretical *velocity* of the developers, taking into account their dedication and the total allocated and unallocated remaining effort, in order to compute the ability to resolve it.
Issues' Due Date compliance	- Accuracy of planning issues' due date	Extracted from the issues tracking available systems (i.e. JIRA and Redmine). This metric is computed as a percentage of the difference between the planned due date of past closed issues and the actual tracked issue closing date.
Delivery Performance	- Timely release delivery - Timely feature specifications delivery - Core component commits - Non-issue component commits	Extracted from the issue tracking system Mantis. Metrics related to timely delivery are computed as percentage of releases/ feature specifications delivered on time in a given period, and total number of releases/ feature specifications delivered in that given period. In case of metrics for tracking commits, it is measured as total number of commits made on components (both core and non-issue/non-core) in a given period close to the delivery deadline.
Blocking (as a factor)	Blocking's factors and metrics [6]	Data source and information to be gathered for Blocking indicator [6]

On-Time Delivery if the past issues' effort has been under/overestimated, and influence the project manager or the responsible person to revise the effort estimation methodology, or to split-up issues in smaller sizes to facilitate more accurate estimations.

Second, *Issues' Development Status* refers to the development status in terms of the planned tasks, the assigned and unassigned effort, the average velocity of the development team, and the remaining time until the release ending date. This factor enables the detection of low productivity, insufficient development resources allocation or high percentage of unassigned effort. For instance, if the factor has low value, it could be fixed with actions to assign more resources to an issue.

Third, *Issues' Due Date Compliance* refers to the percentage of past due dates' compliance. If the percentage is low, it indicates that for some reason, there have been problems in meeting delivery dates in past issues, and the project manager or the responsible should take measures to prevent it from happening again.

Fourth, *Delivery Performance* constitutes metrics that measures a company's adherence to delivery schedules, determined by tracking activities responsible for delivering releases, features, and feature specifications. Here, emphasis is on tracking components of the software product upon which commits were made close to delivery date, which may become potential cause for delays. Consequently, and by focusing on both core and non-core components, delivery schedule compliance and slippage can be estimated.

Finally, *Blocking* [6] refers to the blocking situations that arise when developing. Blocking situations increase waiting time, and, therefore, they are against the flow of constant delivery and can affect on-time delivery of the planned content. The definition of the *Blocking* indicator, including its corresponding factors, (*Feature Definition Completeness, Delayed Tasks, Test Failing, Test Performance, and Low Quality Features)* and metrics may be found in [6].

The presented factors are the ones we identified in the case of at least one industrial partner. In general, when a company wants to define and measure the On-Time Delivery indicator for a specific software product, they can customize the definition of the indicator selecting the factors and metrics that can be computed from the company's available data, and establishes the utility functions and the weights for each metric and factor selected (w_{ij} and w_i) to assess the On-Time Delivery indicator.

V. Related work

As far as we know, there is no definition of On-Time Delivery indicator in the literature, even if it is a crucial aspect in RSD. However, a significant amount of studies has been dedicated to the definition of indicators or variables related to On-Time Delivery, as time-to-market in software development, release readiness, effort estimation and delivery capability in agile software development (ASD). For instance, [7] defines On-Time completion as a dimension of software development performance and identifies related factors. An agile software estimation algorithm is proposed in [8] to make an estimation of the cost, size, and duration of an agile project. Unlike the indicator we propose, On-Time completion and duration of a project are not defined in the context of releases, and factors related to the development status itself are not considered. Staron, Meding and Palm [9] defined an indicator called Release Readiness to predict a product's readiness for deployment in customer environments. The indicator took into account metrics that mainly focus on the testing phase like open defects, defect removal rates, test execution rate, and test pass rate. Furthermore, the indicator was defined in the context of a large mature Agile-Lean project and organization. Factors identified in our paper are a synthesis of different size projects and organizations, and not limited to a specific

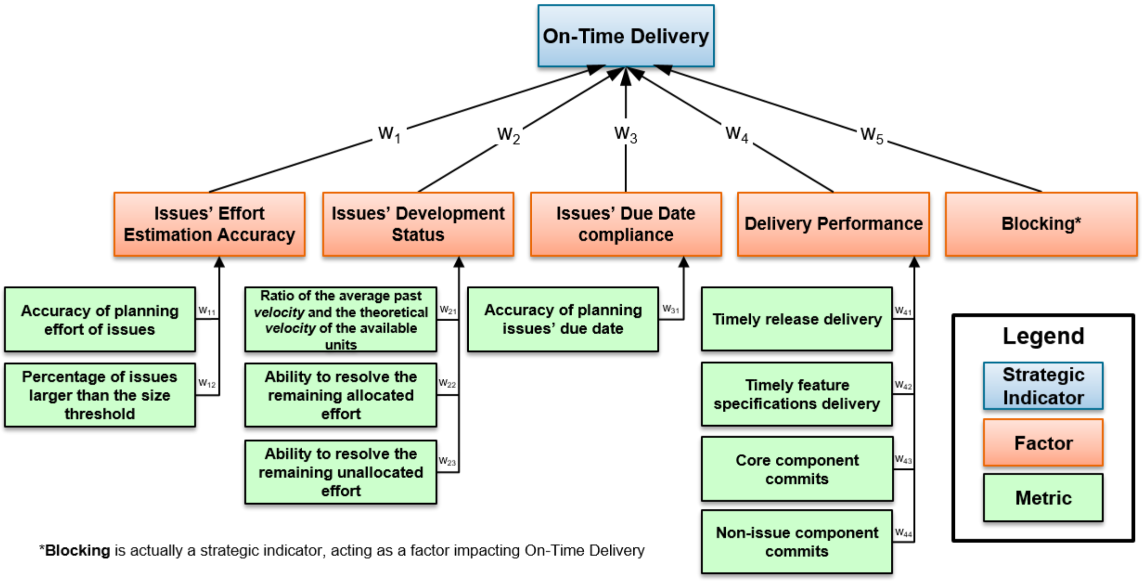
On-Time Delivery
w1
w2
w3
w4
w5
Issues' Effort Estimation Accuracy
Issues' Development Status
Issues' Due Date compliance
Delivery Performance
Blocking*
Accuracy of planning effort of issues
w11
Percentage of issues larger than the size threshold
w12
Ratio of the average past velocity and the theoretical velocity of the available units
w21
Ability to resolve the remaining allocated effort
w22
Ability to resolve the remaining unallocated effort
w23
Accuracy of planning issues' due date
w31
Timely release delivery
w41
Timely feature specifications delivery
w42
Core component commits
w43
Non-issue component commits
w44
Legend
Strategic Indicator
Factor
Metric
*Blocking is actually a strategic indicator, acting as a factor impacting On-Time Delivery

Fig. 2. On-Time Delivery Quality Model

context. In [10] the authors apply data mining techniques to build a delivery capability prediction model on a particular dataset, so their results and extracted variables are linked to the individual datasets used. They do not consider delivery capability as an indicator but as a response variable. A systematic literature review [11] on the use of metrics in ASD highlights the use of burndown charts to keep track of project trends, enabling prediction of completion date. For instance, release burndown helps give a big picture of the entire release, especially if it will be completed on time or slippage can be expected. Similarly, component level burndown helps identify slippage at component level, aimed primarily at resource management. Burndown charts are used mainly to track project progress or to balance workflow. The factors and metrics provided in our paper may complement the information provided by burndown charts, as for example, the additional information about what specific metrics and factors are providing low values for the On-Time Delivery indicator. In [12] and [13] a systematic literature review and a survey of the state of the practice in effort estimation in ASD are presented, respectively, finding that subjective analysis is the most widely used estimation method. Common effort predictors are compiled, being the ones related to the size and complexity of the tasks, and the skills and experience of the team as the ones that were observed in more studies. We do not explicitly include these variables related to the abilities of the team in the On-Time Delivery corresponding factors and metrics, because we have found that they are very hard to estimate individually using the current project management tools (e.g. JIRA, Redmine, etc.). Instead, they are implicitly taken into account with the *Issues' Development Status* factor corresponding metrics: *Ratio of the average past velocity and the theoretical velocity of the available units*, *Ability to resolve the remaining allocated effort* and *Ability to resolve the remaining unallocated effort*.

The main difference between our work and those above mentioned is the fact that we are introducing a generic but customizable way to monitor the On-Time Delivery capability in RSD environments, so any company could adopt and customize it with little effort. The factors and metrics used can be computed from the main project management current tools like JIRA, Redmine, Mantis, or Gitlab.

VI. Conclusions and future work

In this paper, based on industry needs, we have defined the On-Time Delivery indicator as the capability of fulfilling the issues planned for a specific release, and identified Issues' Effort Estimation Accuracy, Issues' Development Status, Issues' Due Date Compliance, Delivery Performance, and blocking situations as the factors that influence the on-time delivery of releases of the software developed.

As part of the future work, we are planning to evaluate the effectiveness of our approach using real data in the context of the Q-Rapids industrial partners, and to iteratively adapt the indicator according to the beliefs of the industrial partners, in order to keep providing an appropriate and meaningful indicator. Moreover, we are going to define a method for assessing the indicator using Bayesian Networks.

Acknowledgment

This work is a result of the Q-Rapids project, which has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement N° 732253. Also, this work is partially funded by the Spanish project GENESIS (TIN2016-79269-R). We thank to Q-Rapids industrial partners for participating in the workshops for defining the On-Time Delivery indicator.

References

[1] L. Guzmán, M. Oriol, P. Rodríguez, X. Franch, A. Jedlitschka, & M. Oivo. "How can quality awareness support rapid software development? - A research preview". REFSQ 2017, 2017.

[2] T. Chow, D.-B. Cao. "A survey of critical success factors in agile software projects". *The Journal of Systems and Software*, 81, pp 961-971. 2008.

[3] S. Martínez-Fernández, A. Jedlitschka, L. Guzmán, A. M. Vollmer. "A Quality Model for Actionable Analytics in Rapid Software Development". CoRR abs/1803.09670 (2018).

[4] D. Avison, F. Lau, M. Myers, & P. A. Nielsen. Action research. *Communications of the ACM*, *42*(1), 1999.

[5] V. R. Basili, "Software modeling and measurement: the Goal/Question/Metric paradigm," Quality. p. 24, 1992.

[6] X. Franch *et al.*, "Data-driven requirements engineering in agile projects: the Q-rapids approach", *REW 2017*, 2017.

[7] G. Lee, W. Xia. "Toward agile: an integrated analysis of quantitative and qualitative field data on software development agility". *MIS Quarterly,* 34, pp 87-114. 2010.

[8] S. Bhalerao, M. Ingle. "Incorporating vital factors in agile estimation through algorithmic method". *International Journal of Computer Science and Applications*, 6, 2009.

[9] Staron, M., Meding, W., & Palm, K, "Release readiness indicator for mature Agile and lean software development projects", International Conference on Agile Software Development (pp. 93-107). Springer, Berlin, Heidelberg.

[10] M. Choetkiertikul, H. Khanh Dam, T. Tran, A. Ghose, J. Grundy. "Predicting Delivery Capability in Iterative Software Development". *Transactions on Software Engineering*, 14(8), 2017.

[11] E. Kupiainen, M. V. Mäntylä, and J. Itkonen, "Using metrics in Agile and Lean software development - A systematic literature review of industrial studies", Inf. Softw. Technol., vol. 62, no. 1, pp. 143–163, 2015.

[12] M. Usman, E. Mendes, and J. Borstler. "Effort estimation in agile software development: a survey on the state of the practice". EASE'15, 2015.

[13] M. Usman, E. Mendes, F. Weidt, and R. Britto. "Effort estimation in agile software development: a systematic literature review". PROMISE'14, 2014.

[14] X. Franch *et al.*, "Data-Driven Elicitation, Assessment and Documentation of Quality Requirements in Agile Software Development," *CAiSE*, pp. 587–602, Jun. 2018.