RELREA - An Analytical Approach for Evaluating Release Readiness

S. M. Shahnewaz  
Department of ECE  
University of Calgary  
Calgary, Alberta, Canada  
smshahne@ucalgary.ca

Guenther Ruhe  
Department of Computer Science,  
Department of ECE  
University of Calgary  
Calgary, Alberta, Canada  
ruhe@ucalgary.ca

Abstract— As part of incremental and iterative software development, decisions about “Is the software product ready to be released at some given release date?” have to be made at the end of each release, sprint or iteration. While this decision is critically important, so far it is largely done either informally or in a simplistic manner, relying on a small set of isolated metrics. In this paper, we present an analytical approach combining the goal-oriented definition of the most relevant readiness metrics with their individual evaluation and their subsequent analytical integration into an aggregated evaluation measure. The applicability of the proposed approach called RELREA is demonstrated for an ongoing public project hosted on GitHub, a web-based hosting service for software development projects. Initial evidence shows that the method is supportive in evaluating release readiness at any point of the development cycle, making projections on the final release readiness and allows determination of bottleneck factors to achieve readiness.

Keywords-release date; release readiness; release criteria; fuzzy set; aggregation; case study

I. INTRODUCTION

To achieve business success of products, in-time and in- quality of releasing software is a key concern of the software companies [1]. Delays in releasing software may cause substantial loss in revenue and in the worst case, failure of the complete software product. One of the main challenges of software product manager is knowing the current state of the readiness of the product based on objectivity [15]. In most cases, release readiness is evaluated at the end of the development cycles (i.e. milestones, sprints, etc.) relying on a set of isolated metrics. This has some major consequences: Firstly, there is no continuity in monitoring product readiness. Secondly, any problem related to release issues cannot be addressed proactively by the development team. Thirdly, if being below expectations in terms of readiness, it is unknown which are the limiting factors.

Evaluating release readiness at any point of time is a challenging task. One of the fundamental issues is that the definition of software release readiness is not well articulated in literature. Most of the related works [2-4, 12] in this domain mainly gauge with defect tracking in order to define release readiness. However, release readiness measures should include all important aspects of the software product [21]. In this paper, we define release readiness as an objective measure, which is determined by aggregating the degree of satisfaction of a set of individual readiness criteria. According to this definition, the main problem of evaluating release readiness is determining the satisfaction levels of the criteria based on monitoring key product and process performance metrics.

In this paper, we propose RELREA, an analytical approach that allows evaluating release readiness at any point in time of the release cycle and making suggestions to improve the status of readiness for the final software product release. The main characteristics of RELREA are:

• Project specific release readiness criteria and metrics are selected in a systematic way.
• Unlike the existing methods (see Section III), overall release readiness is determined by aggregating and evaluating the degree of satisfaction of the criteria.
• Continuous visibility on release readiness status.
• Projection of release readiness at release time.

In the proposed approach, the concept of fuzzy set theory is employed for evaluating the degree of satisfaction of the readiness criteria based on the evaluation of the pre-defined objective measures. As an aggregation operator, Ordered Weighted Average (OWA) is applied. In order to demonstrate the applicability of the proposed approach, an illustrative case-study is presented on an ongoing public project hosted on Github. The initial evidence shows that the new method helps product managers investigating release readiness at any point of time during the development cycles.

The paper is subdivided into seven sections. In Section II, we present the background for concepts used by the proposed method. A brief discussion of related work and tools is done in Section III. The workflow of the proposed analytical approach for evaluating release readiness is presented in Section IV. In Section V, a case-study is presented to demonstrate the applicability of the proposed method. Section VI outlines the discussion on threats to validity. The final section provides conclusions and outlines future work.

II. BACKGROUND

In this section, we provide the fundamental concepts that we have used in the proposed method.
A. Software Release Readiness Criteria

Release readiness criteria can be defined uniformly by considering both organization goals and customer expectation of a software product. Four important dimensions of release readiness (implementation status, testing scope and status, source code quality and documentation status) were identified from existing literature [2-6] and tools [14, 15]. An overview of the release readiness criteria for each dimension is provided in Table I. These factors cover both important artefacts (i.e., features, source code, documents, etc.) and development activities (i.e., coding, building, testing, etc.).

<table>
<thead>
<tr>
<th>Readiness dimensions</th>
<th>Overview of related Release Readiness Criteria</th>
</tr>
</thead>
<tbody>
<tr>
<td>Implementation of functionality</td>
<td>Criteria related to feature implementation, change request implementation, coding effort, continuous integration, build trends, etc.</td>
</tr>
<tr>
<td>Testing</td>
<td>Criteria related to defect finding, defect fixing, reliability, test coverage, test effort, etc.</td>
</tr>
<tr>
<td>Source code quality</td>
<td>Criteria related to code review, coding style, code smells, refactoring, code complexity, etc.</td>
</tr>
<tr>
<td>Documentation</td>
<td>Criteria related to user manual, design documents, test specification, test case documentation, etc.</td>
</tr>
</tbody>
</table>

B. Goal Question Metric (GQM)

The Goal-Question-Metric (GQM) paradigm [8] is the de facto standard to perform software measurement. In its essence, it guides the process of designing an effective measurement program and finding insights from the data collected. In the context of release readiness analysis, our goal is to evaluate the status of the overall readiness. The readiness dimensions corresponding to questions were refining the goal. Available data associated with every question were used to answer them in a quantitative way. An example of defining objective metrics for the testing dimension is shown in Table II.

C. Evaluation of Release Readiness Criteria with the Concept of Fuzzy Set Theory

The concept of fuzzy set theory was introduced by Zadeh [9] in 1965 to represent nonstatistical uncertainty and vagueness associated with data and information. Since then, it has been widely used to solve problems in various decision making environment. In software engineering, fuzzy set theory has been applied successfully to deal with the uncertainty associated with the aspects such as cost estimation [17], reliability prediction [18], and imprecise requirement analysis [19].

Definition 1: A fuzzy set $C$ in a universe of discourse $X$ is characterized by a membership function $\mu_C(x)$ which associates with each element in a real number in the interval $[0, 1]$. The function value $\mu_C(x)$ is termed the “grade of membership” $x$ of in $C$.

The concept of membership function sated in Definition 1, can be used to evaluate the degree of satisfaction of the readiness criteria discussed in II.A. For example, suppose we want to evaluate the satisfaction of defect finding criteria. In this regard, defect arrival rate (DAR) (defects/day) is an objective metric which can be used to evaluate the criteria. Let, $X = R^+ = \{\text{positive real number that can be represented as value of DAR}\}$ and $C = \{\text{satisfaction of defects per day}\}$ be a fuzzy set representing the criteria.

As shown in Figure 1, its membership function is defined as $\mu_C(x)$ which means DAR less than 1 defects/day is considered full satisfaction of defect finding, while defect arrival rate less than 10 defects/day is considered the criteria is not satisfied at all. Values of DAR between, 1 and 10 are considered satisfaction to some degree.

According to this definition, if the value of DAR is 5 defects/day, then the “degree of membership” or “degree of satisfaction of defect finding” will be 0.55. The definition of this membership function is context specific, different projects may have different perception about the satisfaction of defect finding. Thereby, degree of satisfaction of the readiness criteria can be determined by defining the membership functions.

Definition 2: Let, DAR be a real number that can be represented as $x = x_1 + x_2 + \ldots + x_n$ where $x_i$ are real numbers.

As discussed in section II.C and II.D, values of the membership functions represent the degree of satisfaction of
the readiness criteria. We need to aggregate the individual evaluation of the readiness criteria to obtain an overall evaluation. In literature, many aggregation operators are proposed. An overview of the properties of those operators can be found in [11]. Selection of the aggregation operator depends on the context of its use. In the context of evaluating overall release readiness, aggregation operator should have the capability of including the following special considerations along with the general mathematical properties (i.e. boundary conditions, monotonicity, continuity, associativity, symmetry, neutral element, idempotence, etc.).

- The first consideration is the relative importance or weights of the readiness criteria. This allows the decision maker to emphasize important readiness criteria, so that its influence on the overall readiness will be higher.
- Second consideration is the decision maker’s desired decision strategies. For example, a pessimistic decision maker desires that most of the criteria be satisfied. Conversely, an optimistic decision maker will be satisfied if some of the criteria are met.

In this paper, Ordered Weighted Average (OWA) [7] is applied as aggregation operator because of its capability of incorporating the above special considerations. Definition of overall release readiness with OWA operator is stated in the following:

\[
OWA \left( \mu_{C_1}(x), \mu_{C_2}(x), \ldots, \mu_{C_n}(x) \right) = \sum_{i=1}^{n} w_i b_i
\]

Where

i. \( \mu_{C_1}(x), \mu_{C_2}(x), \ldots, \mu_{C_n}(x) \) are the values of the membership functions of the criteria \( C_1, C_2, \ldots, C_n \) with \( \mu_{C_i}(x) \in [0,1] \),

ii. \( w_1, w_2, \ldots, w_n \) are the ordered weights satisfying \( w_i \in [0,1], \sum_{i=1}^{n} w_i = 1 \), and

iii. \( b_i \) is the \( i^{th} \) largest element in the collection \( \mu_{C_1}(x), \mu_{C_2}(x), \ldots, \mu_{C_n}(x) \).

Yager [7] showed that OWA operators lie between maximum ("or") and minimum ("and") of the scores to be aggregated. Details about the properties of OWA operators and ways of including relative importance of the criteria can be found in [20].

IV. RELEASE READINESS APPROACH

In this section, we present the workflow of the proposed RELREA release readiness approach.

A. Workflow of the proposed approach

The workflow of the proposed release readiness analysis method is illustrated in Figure 2. There are 8 main steps:

1) Define Readiness criteria and metrics: At the beginning of the project, based on the GQM approach discussed in Section II.B, the release manager will define the context specific readiness criteria and their corresponding objective metrics.

2) Define membership function: Based on the proposed membership function elicitation techniques described in Section II.D, product manager will define the membership functions by selecting appropriate shapes and their parameters for the readiness criteria.

3) Data collection: At any point of the current development cycle, data related to the pre-defined metrics will be collected (by automated tools).

4) Computation of criteria satisfaction levels: At this point, satisfaction levels of the readiness criteria will be
determined by computing the degree of membership for the current values of the objective metrics at a specific point in time $t_i$.

B. Projection of Release Readiness

Based on the calculated integrated readiness scores over a time period, projected readiness $PR(T)$ for a pre-defined release date $t = T$ is studied. As an approximation $PR(T)$ is calculated based on the gradient of the line between last two evaluation points $E1(t_{i-1}, r_{i-1})$ and $E2(t_i, r_i)$.

$$\text{PR}(T) = \sum w'_i \times \text{PR}(t_i), \text{ for } i=1,...,n$$

In (2), $w'_i$ denotes the weights for each evaluation point. As latest evaluation points are more reliable, more weight is given to the projected scores calculated at later stage of the development compared to the scores of the later stages. This idea is illustrated in Figure 3, where each dashed line points to the projected readiness scores based on last two evaluations.

V. CASE STUDY

A. Context of the Project

Publify\(^1\) is a powerful open source blogging engine hosted on Github. It is one of the oldest Ruby on Rails project started in 2004. Since then, 52 contributors has engaged themselves in this project and contributing continuously to its development. It uses Github\(^2\) for managing issues (i.e. features, bugs, tasks and etc.) and source code versioning. For the continuous integration (CI) it uses the Travis-CI\(^3\) which is a cloud hosted CI platform.

The quality of the source code of the project is tracked with CodeClimate\(^4\) that performs real-time static analysis on the source code repository in order to provide a comprehensive quality report. The latest version (Publify 8.0) of the engine was released on March 01, 2014. Though development activities of this release were kicked off on September 09, 2014, we started to monitor the project from January 26, 2014. Main goal of this case-study is to demonstrate the capability of

\(^{1}\) https://github.com/publify/publify/
\(^{2}\) https://github.com/
\(^{3}\) https://travis-ci.org/publify/publify
\(^{4}\) https://codeclimate.com/github/publify/publify
RELREA in evaluating and predicting release readiness of Publify 8.0 at any point of time during the development.

### Table III. Criteria, Metrics and Data used in the Case-Study: $B_i = \text{values of metrics } M_i$, $X_i, Y_i = \text{parameters of membership functions for which satisfaction level is 0 and 1 respectively. Values of these parameters and relative weights are selected by reviewing the past releases of the project.}$

<table>
<thead>
<tr>
<th>Readiness Dimensions</th>
<th>Readiness Criteria ($C_i$)</th>
<th>Weights</th>
<th>Metric Definition($M_i$)</th>
<th>$B_i$</th>
<th>Parameter of MFs $X_i$</th>
<th>$Y_i$</th>
<th>Satisfaction level (on release date)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Implementation</td>
<td>Satisfaction of feature implementation</td>
<td>0.15</td>
<td>Feature implementation ratio</td>
<td>38.46</td>
<td>10 50</td>
<td>0.71</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Satisfaction of build/continuous integration trends</td>
<td>0.05</td>
<td>Percentage of successful builds/integration</td>
<td>62.34</td>
<td>30 80</td>
<td>0.65</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Satisfaction of implementation effort</td>
<td>0.05</td>
<td>Code Churn per contributor per day</td>
<td>150</td>
<td>50 200</td>
<td>0.67</td>
<td></td>
</tr>
<tr>
<td>Testing (Quality)</td>
<td>Satisfaction of defect finding</td>
<td>0.2</td>
<td>Defect find rate (per day)</td>
<td>0.22</td>
<td>1 0.1</td>
<td>0.86</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Satisfaction of defect fixing</td>
<td>0.1</td>
<td>Percentage of defect fixed</td>
<td>84.21</td>
<td>50 95</td>
<td>0.76</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Satisfaction of defect density</td>
<td>0.08</td>
<td>Defects/KLOC</td>
<td>7.78</td>
<td>12 3</td>
<td>0.47</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Satisfaction of unit test</td>
<td>0.1</td>
<td>Covered LOC/ total LOC</td>
<td>84.58</td>
<td>60 90</td>
<td>0.82</td>
<td></td>
</tr>
<tr>
<td>Source Code Quality</td>
<td>Satisfaction of codes smells</td>
<td>0.08</td>
<td>Number of code smells per class</td>
<td>0.6</td>
<td>1 0.25</td>
<td>0.53</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Satisfaction of code duplication</td>
<td>0.07</td>
<td>Percentage of duplicated code</td>
<td>22.73</td>
<td>40 10</td>
<td>0.58</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Satisfaction of method complexity</td>
<td>0.06</td>
<td>Average method complexity</td>
<td>8</td>
<td>20 5</td>
<td>0.80</td>
<td></td>
</tr>
<tr>
<td>Documentation</td>
<td>Satisfaction of readme file</td>
<td>0.06</td>
<td>Percentage of issues fixed</td>
<td>100</td>
<td>50 100</td>
<td>1.00</td>
<td></td>
</tr>
</tbody>
</table>

### B. Data Collection and Modeling with RELREA

Following the proposed workflow, in step 1 important readiness criteria and metrics were defined from the four dimensions. Details of the selected criteria and metrics are presented in Table III. As the contributors of the project were not available, parameters of the membership function (piecewise linear functions) were defined by studying the values of the metrics from the previous releases.

Relative weights of the criteria were defined by two senior industry practitioners. Required data for calculating values of the metrics were collected from Github, Travis-CI and CodeClimate to determine satisfaction levels of the readiness criteria according to the definition of membership functions. Finally, OWA operator was employed to determine the overall release readiness of Publify 8.0 for three different decision strategies. Further data details can be found in [23].

### C. Release Readiness Analysis of Publify

Using our proposed method, we computed the overall release readiness of Publify 8.0 on six points in time. Figure 4 illustrates the impact of various decision strategies (degree of optimism) in overall release readiness. It shows that PR(170) is highest (0.81) when degree of optimism is considered to be 0.7 and lowest (0.65) when degree of optimism is 0.3. For the optimistic case, more importance is given to the three highly satisfied criteria (satisfaction of user manual (1.0), satisfaction of defect arrival (0.86), satisfaction of unit test coverage (0.82)) while less importance is given to the three low satisfied criteria (satisfaction of implementation effort (0.67), satisfaction of conditions integration (0.65), satisfaction of code duplication (0.58)). Similarly, for the pessimistic case, more importance is given to three low satisfied criteria (satisfaction of defect density (0.47), satisfaction of codes smells (0.53), and satisfaction of code duplication (0.58)) compared to the three highly satisfied criteria (satisfaction of user manual (1.0), satisfaction of defect finding (0.86), satisfaction of method complexity (0.80)).

As a result, product manager can identify the potential trade-off options among the readiness criteria. This kind of analysis of release readiness is helpful when time-to-release is important for the success of a software product. It allows identifying project parameters which need to be adjusted for improving overall release readiness.

![Figure 4. Impact of decision strategies on PR(170).](image)

Figure 5 shows the readiness scores computed by RELREA at different stages of the release cycle as well as the projected readiness scores at $t = 170$. We concluded that achieving a targeted release readiness level of 0.9 is low and that the new version of Publify 8.0 was potentially released too early. In support of this finding, already 15 bugs were reported within 12 days after the release date.

![Figure 5: Calculated and projected readiness PR(170) of Publify 8.0.](image)
VI. THREATS TO VALIDITY

The key threats to the validity of the results are related to
the selection of the release readiness criteria, the applied
membership functions, the metrics selected, and the
aggregation operator applied. As the proposed method allows
defining release readiness criteria uniformly from four
dimensions and metrics are defined in a goal oriented way,
threats to the validity regarding selection process of readiness
attributes are minimized.

One key threat to the construct validity is that the method
only considers the release criteria which can be evaluated
objectively. Subjective release readiness criteria (i.e. testing
depth, user experience, amount of possible enhancements,
etc.) are not considered in the method. However, inclusion of
subjective criteria requires both human experts and more effort
to support continuous evaluation of release readiness.

The accuracy of the calculated release readiness largely
depends on the proper definition of the membership functions.
Consideration of similar past project data and involvement of
the development team will lead to more reliable membership
functions. Selection of proper aggregation operator for
combining the satisfaction levels of criteria is also a threat to
the validity. As we choose a flexible aggregation operator
which can be adjusted by the product release manager, this
threat to the validity is at least reduced.

Another potential threat specific to the case-study is that the
contributors of the projects were not reachable. Thus, we
cannot claim that the selected readiness criteria, relative
weights, metrics and parameters of the membership functions
were truly relevant in the context of the project. However, to
mitigate these issues, we studied the previous releases of the
project and consulted with two senior software engineers for
selecting the values of the required project specific parameters.

VII. CONCLUSION AND FUTURE WORK

Release management is a decision-centric process with a
number of criteria, stakeholders, and constraints involved in it.
We have proposed an analytical approach to evaluate the
release readiness at any point in time of the release interval
and to make projection on its final status. Corrective steps can
be taken to increase the current release readiness level of the
software.

In future, this work will be extended to the analysis of
likelihood of achieving the expected readiness before the
predefined release date. The proposed method needs further
analysis and evaluation of it applicability and usefulness.
Stronger emphasize will be put on tuning of important project
parameters. Integration of the framework with the existing
project management tools such as JIRA, Team Foundation
Server (TFS), and Github is intended to achieve an increase of
its acceptability for industry practitioners.

ACKNOWLEDGMENT

This research was partially supported by the Natural
Sciences and Engineering Research Council of Canada,
NSERC Discovery Grant 250343-12. Discussions with Trong
Tan Ho, S. M. Didar-Al Alam, and Muhammad Rezaul
Karim were helpful in conducting the project case study.

REFERENCES

Metrics," in IEEE Conference on Soft Computing in Industrial
[5] A. Asthana and J. Olivieri, "Quantifying Software Reliability and
Readiness," in IEEE International Workshop Technical Committee on
Communications Quality and Reliability; 2009, pp. 1-6.
across the software development life cycle " Department of Computer
Science & Engineering,University of California, Riverside 2013.
in multi-criteria decision making," IEEE Transactions on Systems, Man
Question Metric Approach". In Encyclopedia of Software Engineering.
1965.
Theoretical and empirical work," Fundamentals of fuzzy sets, pp. 195-
initiative in Soft Computing, Computer Science Division: University of California,
[12] T. Pearse, T. Freeman, and P. Oman, "Using Metrics to Manage the End-
Mature Agile and Lean Software Development Projects," Agile
[15] "PTC integrity: Application Lifecycle Management (ALM)," Last
accessed March 14, 2014. in http://www.ptc.com/solutions/application-
lifecycle-management/.
[16] O. S. Yamada, "Optimum Software-Release Time Considering an Error-
Detection Phenomenon During Operation," IEEE Trans. on Reliability,
estimation with fuzzy models." ACM SIGAPP Applied Computing
of software reliability prediction," in Conference on Software Testing,
Reliability and Quality Assurance, 1994, pp. 154-158.
logic-based requirements analysis," European Journal of Operational
process using OWA operators," Journal of Intelligent and Fuzzy
[21] Jrothman, "Release Criteria: Is This Software Done?," Last accessed
March 14, 2014. in http://www.jrothman.com/2002/03/release-criteria-
is-this-software-done/.
https://sites.google.com/site/shawniute/release-readiness