Towards the Improvement of Use Case Models:The AIRDoc Process

Ricardo Ramos [1],
Jaelson Castro [2]
[1] CECOMP, Universidade Federal do Vale do São Francisco (UNIVASF)
Juazeiro, BA - Brazil
[2] CIn, Universidade Federal de Pernambuco (UFPE)
Recife, PE - Brazil
ricargentonramos@gmail.com, jbc@cin.ufpe.br

João Araújo [3],
[3] Universidade Nova de Lisboa (UNL)
Caparica, Lisboa - Portugall
ja@di.fct.unl.pt

Fernanda Alencar [4]
[4] DES , Universidade Federal de Pernambuco (UFPE)
Recife, PE - Brazil
fernanda.ralencar@ufpe.br

ABSTRACT

A requirements document tends to be full of requirements that are no longer meaningful, descriptions that are unnecessarily long-winded and complicated, and duplicated information, among other shortcomings. These problems hinder the overall understandability of the requirements document and might compromise the subsequent development phases. A requirements document quality evaluation strategy should incorporate also guidelines to help practitioners to effectively measure and improve their requirements documents. In this poster paper we brief describe a process that supports the evaluation and improvement of use case models named AIRDoc.

Categories and Subject Descriptors

D.2.1 [**Software Engineering**]: Requirements/Specifications – languages, methodologies.

General Terms

Measurement, Documentation, Languages.

Keywords

Use Case Model; refactoring, Goal Question Metrics.

1. INTRODUCTION

The definition of requirements is critical for the success of software system development. However, current practices often lead to inaccurate and faulty requirements descriptions.

Requirements documents may come in different styles, for example, some advocate the use of goal oriented modeling languages such as i* [1] and KAOS [2]. Those developing critical and safe applications may prefer a more formal approach. Nevertheless, the great majority of requirements documents are written either in natural language or in some semi-structured notation. This work focuses on Use Case models. Besides being part of an international standard [9], it is also very popular in our subject domain, i.e., the Brazilian software industry.

Some attention has been dedicated to discover typical syntactical problems that compromise the quality of Use Case Models, such as requirements that have been abandoned and that are no longer meaningful, descriptions that are too long and difficult to read, and information that is duplicated [3][4]. These shortcomings hinder the overall understandability and reusability of Use Case Models throughout the development process. We believe that some of the potential Use Case problems could be solved using suitable refactorings [5][6], at the appropriate time and place. We claim that the early identification of symptoms and the removal of their causes can greatly improve the quality of Use Case Models, contributing to keep the software development within budget and time.

Hence, we describe AIRDoc, an approach to improve the quality of Use Case Models. Thus, our objective is twofold:

- To evaluate Use Case Models by elaborating goals, defining questions and assertions that will be addressed by metrics;
- As well as to improve the quality of Use Case Models by means of refactorings. AIRDoc relies on a catalog of well-known requirements problems and their possible solutions.

2. AIRDoc Process

AIRDoc is an acronym for “Approach to Improve Requirements Documents”. The AIRDoc process is based on Goal Question Metrics GQM [7] and complies with the IEEE Standard for a Software Quality Methodology [8]. It is divided into two stages:

(1) Evaluation that consists of four Activities, which are composed by steps or other activities (sub-process):

(E.1) Elaboration of Evaluation Plan;

(E.2) Definition of GQM Activities,

(E.3) Collection of the Metrics Values;

(E.4) Interpretation of GQM Activities;

(2) Improvement that consists of two Activities:

(I.1) Elaboration of Improvement Plan; and

(I.2) Perform Improvement.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
SAC'11, March 21-25, 2011, TaiChung, Taiwan.
Copyright 2011 ACM 978-1-4503-0113-8/11/03…$10.00.

The quality team decides where the process should start. Hence, the first stage, the "evaluation phase", is optional. For example, when some potential requirements problems (such as duplicated steps or large use cases) have already been identified. However, if the existence of problems is already known, but their localization (i.e. the exact place) cannot be determined, it is advisable to run the evaluation phase (which requires more careful planning and appropriate budget and time). In doing so, other problems that could have gone unnoticed may also be identified.

In the sequel, we overview the six activities of the two stages of the AIRDoc process.

E.1 - Elaboration of Evaluation Plan - The first stage of our approach is the evaluation of the use case model, which consists of 4 activities (see Figure 1).

E.1.1 - Definition of Quality Team – In this first activity the team members are selected;

E.1.2 - Selection of Tools and/or other Resources – This second activity defines the tools or/and resources to be used in the evaluation phase;

E.1.3 - Definition of Software Quality Requirements – This third activity establishes the quality issues to be evaluated.

E.1.3.1 - Establish the Quality Evaluation Scope - This sub-activity describes the necessary information to understand the scope of what will be measured.

E.1.3.2. Select the Quality Attributes of the Evaluation – The second sub-activity defines the quality attributes of interest.

E.1.3.3. Define the Evaluation Goal – The third sub-activity defines the objective of the evaluation.

E.1.4 - Generation of Project Plan - The fourth and last activity of the first stage of our evaluation approach defines the plan to be followed.

E.2 - Definition of GQM Activities - The second activity of the evaluation phase is divided into 4 sub-activities, the first one (E.2.1) deals with the selection and definition of a quality model that will be the base for evaluation, the following two (E.2.2 and E.2.3) define questions and metrics used in the GQM approach and the last one (E.2.4) addresses the elaboration of assertions that will assist the quality team to interpret the metrics values.

E.3 - Collection of the Metrics Values - The activities of this it are to ensure the quality of the collection of the metrics and to avoid mistakes.

E.4 - Interpretation of GQM Activities - The measurement values are used either to accept or reject assertions.

I.1 - Elaboration of Improvement Plan - All problems, or symptoms, detected are discussed and plans for improvement are proposed. This activity is composed of two steps:

I.1.1 - Problem Analysis - This is related to the identification of a problem type. To be able to select the appropriate improvement, it is necessary to analyze the problem(s) detected. The AIRDoc approach includes a catalog of potential problems that could be used in this step.

I.1.2 - Definition of the Solution - Having identified the problem, the next step of the Elaboration of the Improvement Plan (I.1) is to define how to apply the solution. After selecting the solution(s) (refactorings), the quality assurance team needs to document and plan how to apply the solution(s).

The AIRDoc provides a catalog of Refactoring [10][11]. This catalog describes a set of solutions (refactorings). The quality assurance team needs to create a plan to apply the selected solution. This plan contains the tasks and who will execute it.

I.2 - Perform Improvement - All the refactorings selected are applied in the Use Case Model and the new model are evaluated.

3. Conclusions

The low quality of requirements documents cause defects that may be propagated to later phases, where the cost of fixing such defects escalates significantly. In this poster paper, we presented the AIRDoc process to evaluate and improve requirements documents specified by Use Case Models.

It is worth noting that AIRDoc is not appropriate to find semantics problems. However, syntactic problems are abundant in requirements documents and also need to tackled.

4. REFERENCES

[1] E. Yu, "Modeling Strategic Relationships for Process Reengineering,“ Ph.D. thesis, Department of Computer Science, University of Toronto, Canada, 1995.

[2] A. Dardenne, A. van Lamsweerde, and S.Fickas, "Goal–directed Requirements Acquisition" Science of Comp. Programming, 1993.

[3] S. Lilly, "Use Cases Pitfalls: Top 10 Problems from Real Projects Using Use Cases". Proc. TOOLS USA '99, IEEE C. S., 1999.

[4] A. Cockburn, Writing Effective Use Cases. Addison-Wesley, 2000.

[5] W. Yu, J. Li, and G. Butler, "Refactoring use case models on episodes," Proc. of the 19th IEEE International Conference on Automated Software Engineering, (ASE 2004). Linz, Austria, 2004.

[6] G. Overgaard, and K. Palmkvist, Use Cases Patterns and Blueprints. Addison Wesley Professional, 2004.

[7] V. R. Basili, G. Caldiera, and H. D Rombach, "The goal question metric approach. " Encyclopedia of Software Engineering, 1994.

[8] IEEE Standard for a Software Quality Metrics Methodology, IEEE Std. 1061-1992, 1992.

[9] UML Resource page, OMG - Object Management Group, http://www.uml.org, 2010.

[10] R. Ramos, E. Piveta, J. Castro, J. Araújo, A. Moreira, P. Guerreiro, M.Pimenta, and R. Tom Price. Improving the Quality of Requirements with Refactoring. In: VI Brazilian Symposium on Software Quality, (SBQS2007), Recife, PE, Brazil, 2007, pp. 141-155.

[11] R. A. Ramos, J. Araújo, A. Moreira, J. Castro, F. Alencar, and R. Penteado. Early Aspects Requirements. Journal of IEEE Latin América, v. 6, 2008, pp. 238-243.