PROMETEU - a tool to support documents generation and traceability in the test process.

Jorge Luiz da Cruz, Mario Jino (Orientador), Adalberto Crespo (Co-orientador)

Departamento de Engenharia de Computação e Automação Industrial (DCA)
Faculdade de Engenharia Elétrica e de Computação (FEEC)
Universidade Estadual de Campinas (Unicamp)
Caixa Postal 6101, CEP 13083-970 – Campinas, SP, Brasil

prof.jorgecruz@gmail.com; jino@dca.fee.unicamp.br; adalberto.crespo@cenpra.gov.br

Abstract – Software testing must be well documented and based on up to date and consistent information to avoid troubles such as incomplete or inappropriate tests. This involves the definition of what must be recorded, beyond tracing the diverse links among the many information elements that compose the documents. Documenting and tracing involve the manipulation of a very huge amount of data, demanding automated support. This paper presents PROMETEU, a tool to support documents generation and traceability among artifacts that comprises the test process documents.

Keywords – Software testing; testing documents; traceability; software testing traceability; IEEE Std 829-1998.

1. Introduction

The software testing management and execution must be based on up to date, consistent and precise information during the whole process [1], [6]. This involves the definition of what must be recorded, beyond the creation and maintenance of many traceability links among information elements that compose documents [3], [7]. The amount of data manipulation is huge, demanding automated support.

This paper presents PROMETEU, a tool to support documents generation and traceability among artifacts and information elements that comprises the test process documents. The documents are based on IEEE Std 829-1998 Standard. Artifacts are all relevant information produced, recorded and modified as part of a software process. An information element is a piece of an artifact (e.g. a document section). The rest of this paper is organized as follow: Section 2 briefly presents traceability; Section 3 presents the IEEE Std 829-1998 Standard; Section 4 briefly describes the tool PROMETEU, and Section 5 presents the conclusions.

2. Traceability

Traceability is an important characteristic of good quality software [7], [3]. Traceability can be defined as: “the ability to describe and follow the life of a requirement, in both a forward and backward direction, i.e., from its origins, through its development and specification, to its subsequent deployment and use, and through periods of ongoing refinement and iteration in any of these phases” [3]. Two broad definitions: (a) "The degree to which a relationship can be established between two or more products of the development process, especially products having a predecessor-successor or master-subordinate relationship to one another"; (b) "The degree to which each element in a software development product establishes its reason for existing " [4].

2.1. Traceability modeling

Implementing traceability demands analyzing its use context to define which artifacts and information elements must be traced, what type of links must be established among artifacts [3], [6], and the main type of user [7].

2.2. Traceability in the test process

Traceability can improve the test process in many ways [1], [6]: enabling requirements-based testing; keeping documents updated and consistent; enabling early revision of requirements; improvement in change impact analysis, easily discovering of artifacts and process players related to a software defect, etc.

3. Test process documents

A set of documents carefully designed and based on up to date and consistent information are crucial to achieve success in a testing process. The visibility of the process is increased, contributing effectively for its management [6].

The IEEE Std 829 Standard can be used to guide the definition of the set of documents required by a test process [5]. The eight Standard documents are: a) Test Plan; b) Test Design Specification (TDS); c) Test Procedure Specification (TPS); d) Test Case Specification (TCS); e) Test Log; f) Test Incident Report (TIR); g) Test Summary Report; h) Test Item

Transmittal Report. Many projects, however, can get by with a smaller number of documents [2].

4. The tool PROMETEU

PROMETEU is a prototype tool that was developed in a context of a methodology aimed to introduce or improve the software testing process of small software companies [2]. Its main features are: Test documents based on IEEE Std 829 standard; Standardized and parameterized documents generation; Software requirements recording; Traceability among artifacts, information elements and process players; Record and management of test cases execution; Test incidents record and control; Process status reports.

The artifacts and information elements to trace are based on IEEE Std 829 Standard. The V Model corroborates the use of artifact types like software requirements and designs [1]. Additional artifacts links maintained are based on the work of Gotel and Finkelstein [3] and Ramesh and Jarke [7]. The link types are based in the existing relations between the information elements of IEEE Std 829 documents and the work of Gotel and Finkelstein [3], Pfleeger and Bohner [6] e Ramesh and Jarke [7].

The artifacts managed by the tool are: a) process players [3], [7]; b) software requirements [1]; c) software designs [1]; d) test documents [5]. Considering the focus on low-end users the traceability links and the visualization ways are predefined. The links don't have semantic attributes, and they are recorded and retrieved like tables and relations of a relational database. There is horizontal traceability among Software Requirements, Software Designs, TCS, Test Plans, TDS, TPS and TIR. There is vertical traceability among Software Requirements and among Test Case Specifications.

Concerning traceability information, every artifact type has a set of attributes to trace, for example: associated process rationales [3], [7], associated process participants, associated comments and assumptions, and when a change was made.

4.1. Other tools

Traceability is present in commercial tools. For instance, IBM Rational tools also support traceability among software objects: RequisitePRO supports traceability among only requirement documents; TestManager links test plans and test cases; ClearQuest manages only defect documents. PROMETEU supports additional object types leveraging the benefits of traceability. Besides, PROMETEU is based on an international standard [5].

4.2. Results

Two papers were published until now and some organizations are interested in the tool. Also, as suggested by many people, PROMETEU can be used as an educational tool to teach Software Testing discipline.

5. Conclusions

PROMETEU was developed to support traceability among the artifacts that comprise documents of the test process. Its primary objective is to foster the generation and maintenance of these artifacts. The tool was designed to make easier to obtain a well documented and based on up to date and consistent information testing process. The artifacts and information elements handled were mainly based on the IEEE Std 829 Standard. Future improvements include traceability among many test cycles, process activities and tasks, as well as some changes on graphical interface to make easier visualizing traceability links. PROMETEU is a test document management tool, not a requirements management tool.

6. References

[1] Craig, R. D. and Jaskiel, S. P., Systematic Software Testing, Artech House, 2002.

[2] Crespo, A. N. et. al., "Application of the IEEE 829 Standard as a Basis for Structuring the Testing Process", The Journal of Software Testing Professionals, Vol. 3, No. 3, December 2002.

[3] Gotel, O. and Finkelstein, A., "Contribution Structures", Proceedings of 2nd International Symposium on Requirements Engineering. IEEE Computer Society Press, 1995, p. 100-107.

[4] IEEE Std. 610, "IEEE Std 610: Standard Computer Dictionary", Institute of Electrical and Electronic Engineers, USA, 1991.

[5] IEEE Std. 829, "IEEE Std 829: Standard for Software Test Documentation", Institute of Electrical and Electronic Engineers, USA, 1998.

[6] Pfleeger, S. L., Software Engineering: Theory and Practice, Second Edition. Prentice Hall Inc, 2001.

[7] Ramesh, B. and Jarke, M., "Toward reference models for requirements traceability", IEEE Transactions on Software Engineering, Vol. 27, Issue 1, January 2001, pp. 58 – 93.