Using Scenarios to Derive Software

Graciela Dora Susana Hadad
Thesis Advisor: Prof. Dr. Julio Cesar Sampaio do Prado Leite
Co-Advisor: Prof. Ing. Jorge Horacio Doorn
Universidad Nacional de La Plata, Facultad de Ciencias Exactas, October 2008
Doctoral Thesis
gracielahadad@gmail.com

The present thesis describes a Requirements Engineering strategy, called SDRES (Scenario Driven Requirements Engineering Strategy), which proposes to face topics rarely treated in real practice, such as continuous changes in requirements, software defects originated during requirements production, the organisational context surrounding the software system and the treatment of quality requirements.

Several statistical reports on software project failures, such as Chaos Reports and GAO Reports, done during last decades established that requirements are the main source of problems in software projects. Besides, Software Engineering has paid more attention to model and implement products than to examine aspects related to the requirements definition and its dependency on social aspects that involve software development and operation. Therefore, Requirements Engineering must attend to non-technical aspects: i) to establish a good communication among stakeholders, and ii) to deeply understand the user world.

Another obstacle in software development is requirements evolution. Continuous changes in software are motivated in a great spectrum of causes that frequently should be attended at once. The nature of these software changes is highly related to changes in requirements.

Thus, the key points that a requirements process should take into account are:

i. Involve clients and users in the development process. This requires the achievement of a perdurable and active participation of them. A good communication is essential to understand their necessities.
ii. Assure requirements quality. The use of adequate V&V techniques as continuous activities through the requirements process is mandatory.
iii. Treat requirements evolution as a perennial situation. The development process should be aware of evolution and manage it. Therefore, requirements management with a traceability mechanism should be implemented.
iv. Consider social, organizational, political and legal aspects of the context where the software will operate. The use of organizational modeling techniques and appropriate elicitation techniques is fundamental to understand the application domain.
v. Tend to requirements completeness. This issue is related to quality aspects and to the domain knowledge acquisition.

This strategy is oriented to the client and driven by models: Language Extended Lexicon, Scenarios and Software Requirements Specification. Those models are written in natural language to facilitate communication and elicitation. SDRES keeps in mind the quality of the models providing verification and validation processes. For each activity of the strategy a set of heuristic and recommendations is presented. The evolution topic and the model versioning are treated, as well as different modalities to apply the strategy according to several aspects: the complexity of the problem, the previous knowledge on the application domain, and other domain and project characteristics.

SDRES aims to obtain a better comprehension of the problem in order to create better negotiated solutions. It establishes a solid bridge between customer necessities and designer solutions providing good domain understandability and an anchor on domain vocabulary.

SDRES involves all the requirements engineering activities: elicitation, modeling, analysis and management, and consists basically in:

i. Understanding the vocabulary used in the universe of discourse, help by the Language Extended Lexicon model;
ii. Understanding the current universe of discourse, help by a set of current scenarios that represent the situations observed in the application domain;
iii. Defining the software system context, by producing a set of future scenarios that represent situations envisioned in a future application domain where the software system will be operating; and
iv. Making explicit the requirements, by producing a SRS where the requirements are clearly individualized from the future scenarios set.

SDRES gives a practical solution to the following topics:

Communication

- ✓ A glossary with universe of discourse terminology is created (Language Extended Lexicon).
- ✓ Natural language-based models and any project documentation could be written using the universe of discourse terminology.

Evolution

- ✓ Scenario models help to envision, estimate and execute requirements changes.
- ✓ Software Requirements Specifications are a tool for the software process management, provide a

checklist to verify the design and provide a checklist for software testing.

Traceability

- ✓ The models with hypertext links enhance pre and post navigation.
- ✓ A traceability model supported by a stack of traces virtually ties actual UofD needs to software system.

The assurance of completeness within semi-formal strategies is extremely difficult. This is also true with SDRES. However, significant efforts have been made to be exhaustive in minimizing this particular drawback throughout the entire process. The key practices are:

i. The Language Extended Lexicon is consistently improved along the process.
ii. The integration of scenarios producing high level scenarios may suggest semantic gaps among scenarios. This operation is applied both to current scenarios and future scenarios.
iii. The building of future scenarios by means of combining software system goal and current scenarios widens the elicitation focus.
iv. The activities of verification and validation of the requirements models promote a quality oriented approach.

Recording software requirements by means of natural language structures is highly attractive since this allows the participation of all the agents involved in the development process, avoiding the classical knowledge barriers that arise when using other representation schemas. It has been observed that the use of slightly formalized natural language structures, such as the language extended lexicon and scenarios, does not introduce severe difficulties to non specialized software development people when reading them.

Scenarios allow modeling not only the current and future software system characteristics but also the current and future behavior of the universe of discourse. Future scenarios enable the possibility of viewing influences of the software system upon the expected context itself. SDRES aims to obtain a better comprehension of the problem in order to create better negotiated solutions. It establishes a bridge between customer necessities and designer solutions.