

Software Requirements Patterns

A State of the Art and the Practice

Xavier Franch

Group of Software and Service Engineering (GESSI)
Universitat Politècnica de Catalunya (UPC-BarcelonaTech)
Barcelona, Spain
franch@essi.upc.edu, <http://www.essi.upc.edu/~franch/>

Abstract—Software requirement patterns are an increasingly popular approach to knowledge reuse in the requirements engineering phase. Several research proposals have been formulated in the last years, and this technical briefing presents them. Beyond that, a report on the current adoption of these proposals (or any other ad-hoc approach) in industry is presented. This state of the practice will show that the need to pave the road to successful adoption still persists.

Index Terms— Software Requirements Patterns, Requirements Engineering, Patterns, Requirements Reuse, Knowledge Engineering, Empirical Study, Literature Review, Survey.

I. INTRODUCTION

Originally conceived for code, reuse has become a central activity in all software development related processes. Requirements engineering (RE) is not an exception to this rule [1]. The reuse of software requirements may help requirement engineers to elicit, validate and document software requirements and thus obtain software specifications of better quality through more effective processes [2]. The use of patterns is a consolidated approach to reuse that has gained increasing attention in RE, both in industry and in academy.

This technical briefing taught at ICSE 2015 presents a state of the art and state of the practice on software requirements patterns (SRP). The contents is based in classical RE textbooks and is complemented with the results of several recent empirical studies: case studies on the use of SRP in different contexts (e.g. call-for-tender processes); exploratory surveys based on e-questionnaires responded by RE practitioners; and an updated systematic literature review not just on patterns but on requirements reuse. All these studies are consolidated to present a unified view.

The briefing is addressed to researchers, practitioners and educators in software engineering, especially requirements engineers. For researchers, an updated state of the art is exposed. For practitioners, processes and templates are outlined and a successful case study of pattern-based RE is presented. For educators, the briefing provides the basis for developing course material.

II. BACKGROUND

Recent reports show how still today a significant percentage of software projects are out of budget, suffer delays or simply have to be cancelled. One of the most recognized

reason for this scenario is the failure in producing a good set of software requirements [3]. Several studies have explored this link throughout the last decades. As far as in 1981, Boehm already mentioned that approximately 60% of all errors in system development projects originate during the RE phase [4]. Also the high cost of fixing requirement errors was stated in that work. Even if the RE field has substantially evolved since then, still it causes problems in industrial projects. A recent Gartner report [5] states that requirements defects are the third source of product defects (following coding and design), but are the first source of delivered defects (in particular for service projects). Other up-to-date studies [6][7] report that the definition of requirements remains a grand challenge for information technology professionals. Methods for improving the quality of software requirement specifications are therefore still needed.

From the several possible approaches to reuse in RE, we are interested in the concept of pattern [8]. Software engineering practitioners have adopted patterns in several contexts, remarkably related with software design (e.g., design patterns and software architectural patterns), but also in other software development phases, both earlier and later. In this briefing, we provide insights in the use of patterns in the RE phase, namely Software Requirement Patterns (SRP).

An SRP is a guide for writing a particular type of requirement [9]. There are several perspectives that are fundamental in a proposal of SRP which are presented in the technical briefing: theoretical (ontologies, metamodels), methodological (processes around) and organizational (impact, cost analysis).

III. OBJECTIVES

The overall goal of the technical briefing can be stated as: having an overview of the state of the art and the state of the practice in SRP definition and adoption. Of course, this is an ambitious goal that could be subject of a larger tutorial. Therefore, it is important to understand that the briefing is more broad than deep. To make an analogy with well-known empirical terms applied to scientific literature analysis, it is closer to a systematic mapping than to a systematic literature review: we want to know what are the main research and practice streams of action, instead of investigating some particular topic in depth.

Some more concrete objectives are:

- Understanding the concept of SRP and the different approaches that exist, articulated around several dimensions.
- Analyze in more depth one particular approach (by Withall [9]) for illustration purposes.
- Understanding the processes around SRP: construction and evolution of catalogue of SRP, and application in a particular project [10][11].
- Having a look to the general structure and contents of particular catalogue of SRP as a way to better understand the practical implications of the concept [12].
- Knowing the current practices of the community by the results of existing empirical studies ran with practitioners as population [13][14].
- Learning the insights and lessons learned of a large-scale case study of construction of an SRP catalogue.

IV. CONTENTS

The briefing is structured as follows:

A. Part I. Preliminaries

A motivation aligned with the first two sections of this paper opens the briefing. Fundamentals of SRP are provided basically following the seminal textbook by Withall [9].

B. Part II. State of the art

In a literature review, we found several approaches to SRP that may be classified along different dimensions: (1) the activity where they are used (e.g., requirement elicitation and documentation, knowledge management); (2) their scope: domain-dependent (e.g., embedded systems) or generic (e.g., Withall's catalogue); (3) the language: they may be expressed in natural language or some other formal or at least rigorous notation; (4) the type of project: a particular type (e.g., call for tender processes) or not; (5) their structure: it may be informally defined or compliant to some grammar or metamodel. The need for classification schemas to organize the contents of the catalogue is also emphasised and some existing approaches surveyed.

C. Part III. State of the practice

This part will present results from three types of empirical studies that addressed empirical studies on requirements reuse have been found: case studies [15]; surveys and interviews on requirements engineering in general [16]; surveys and interviews on requirements reuse in particular [13][14].

D. Part IV. Conclusions

The briefing ends with a summary and provides some open questions and future research directions.

V. CONCLUSIONS

This briefing provides the basis for understanding the SRP concept and their adoption in practice. Said that, it is clear that every IT organization will have its own peculiarities which will lead to different instantiations of the generic concept and

processes. Small organizations may opt by more lightweight solutions, whilst big companies that handle dozens of projects a year with lots of similarities may benefit from a more rigorous and staged approach.

ACKNOWLEDGMENT

This work is partially supported by the Spanish project EOSSAC, ref. TIN2013-44641-P.

REFERENCES

- [1] W. Lam, J. A. McDermid, A. J. Vickers. "Ten Steps towards Systematic Requirements Reuse". *Requirements Engineering Journal* 2(2), 1997.
- [2] S. Roberson, J. Robertson. *Mastering the Requirements Process* (3rd ed.). Addison-Wesley, 2012.
- [3] H. Hofmann, F. Lehner. "Requirements Engineering as a Success Factor in Software Projects". *IEEE Software*, 18(4), 2001.
- [4] B. Boehm. *Software Engineering Economics*. Prentice Hall, 1981.
- [5] Gartner Group. *Hype Cycle for Application Development: Requirements Elicitation and Simulation*, July 2011.
- [6] SwissQ Consulting. *SwissQ Requirements Trends & Benchmarks Switzerland*, 2012.
- [7] R. Young. "The Main Thing is Keeping the Main Thing the Main Thing". *Requirements Engineering Magazine*, 2014-02, IREB.
- [8] C. Alexander. *The Timeless Way of Building*. Oxford Books, 1979.
- [9] S. Withall. *Software Requirements Patterns*. Microsoft Press, 2007.
- [10] S. Renault et al. "A Pattern-based Method for building Requirements Documents in Call-for-tender Processes". *International Journal of Computer Science & Applications* 6(5), 2009.
- [11] X. Franch, C. Palomares, C. Quer, S. Renault, F. de Lazzar. "A Metamodel for Software Requirement Patterns". In *Proceedings of the 16th International Working Conference on Requirements Engineering: Foundation for Software Quality (REFSQ)*, 2010.
- [12] X. Franch, C. Quer, S. Renault, C. Guerlain, C. Palomares. "Constructing and Using Software Requirement Patterns". In *Managing Requirements Knowledge*, Springer, 2013.
- [13] C. Palomares, X. Franch, C. Quer. "Requirements Reuse and Patterns: A Survey". *Proceedings of the 20th International Working Conference on Requirements Engineering: Foundation for Software Quality (REFSQ)*, 2014.
- [14] A. Hoffmann, A. Janzen, H. Hoffmann, J.M. Leimeister. "Success Factors for Requirement Patterns Approaches Exploring Requirements Analysts' Opinions and Whishes". In *Sozio-technisches Systemdesign im Zeitalter des Ubiquitous Computing (SUBICO) im Rahmen der Informatik*, 2013.
- [15] M. Eriksson, J. Börstler, K. Borg. "Managing Requirements Specifications for Product Lines—An Approach and Industry Case Study". *Journal of Systems and Software*, 82(3), 2009.
- [16] M. Niazi, K. Cox, J. Verner. "A Measurement Framework for assessing the Maturity of Requirements Engineering Process". *Software Quality Journal*, 16(2), 2008.