

Chapter 9

Getting More Out of Software Prototypes: A Systematic Literature Review

Alejandro Sánchez-Villarin

University of Seville, Spain

Alejandro E. Santos

University of Seville, Spain


José E. Gonzalez-Enriquez

University of Seville, Spain

Javier J. Gutierrez

University of Seville, Spain

Nicolas E. Sanchez-Gomez

 <https://orcid.org/0000-0001-9102-6836>

University of Seville, Spain

Nora E. Koch

University of Seville, Spain

Maria Jose Escalona

 <https://orcid.org/0000-0002-6435-1497>

University of Seville, Spain

ABSTRACT

Prototyping is a requirement technique frequently used for communication between customers and developers. Software prototypes help to understand users' expectations, but they are often seen as disposable artifacts because it is not easy to manage the transfer of knowledge from prototypes to software models or code. This chapter studies whether suitable solutions already exist for exploiting the knowledge acquired during the building of prototypes in the early phases of the lifecycle. The objective is addressed by means of a systematic literature review of approaches offering solutions for transforming software prototypes into analysis models. We propose a characterization schema for comparing them and describing the current state-of-the-art. The results reveal a need for more automated solutions that are more economical in terms of time and effort for transforming prototypes into models and thereby ensuring traceability between requirements and design artifacts.

INTRODUCTION

Prototypes can help to address what is unquestionably one of the most critical aspects of software engineering: requirements engineering, or the correct definition of end user¹ needs (Fernández et al., 2017).

DOI: 10.4018/978-1-6684-9576-6.ch009

The efficient acquisition, definition, validation, and maintenance of such needs are key to all software development and future maintenance (Escalona & Aragón, 2008).

Despite all the research and the importance that has been attached to requirements, this topic still has much room for improvement and has been the object of many studies (Fernández et al., 2017). Some works describe the development of different techniques. One of the most effective techniques for understanding the user has been found to be software prototyping (Canedo et al., 2020). In other engineering environments, such as aerospace or industry, the use of prototypes is not new (Jensen et al., 2016). This can clearly be seen in methodologies and references like Design Thinking (Plattner et al., 2009), Human Computer Interaction mechanisms (Yıldız, 2019) and User-centered Design (Dhandapani, 2016). In software development, however, prototyping is a somewhat controversial issue (Ali, 2017).

Prototype development usually facilitates user communication and is therefore often used precisely for that purpose (Beaudouin-Lafon & Mackay, 2003). In many cases, however, prototypes are discarded very early in the development process because work done on them is deemed to be “wasted” in the analysis and software design phase (Kuczenski et al., 2021). Proof of this are prototypes that are never looked at again, but just saved in repositories of old projects. These represent a waste of resources that could even be re-used for similar projects (Nguyen-Duc et al., 2017).

This paper presents a study into the use of prototypes in the software development process, both in research and in industry. It also looks at solutions that allow the effective use of prototypes (knowledge provided by prototypes) and analyzes problems associated with this practice. We call this technique “reusing” prototypes.

In industry, where it is necessary to reduce project costs, prototyping tends to be seen as a negligible activity. This should not surprise us. In many cases, prototypes are produced in external environments (mockup or interface design tools) which, once validated, are discarded and the information passed to development environments. In other words, a team develops and validates the prototypes and the information collected is then manually passed to analysis or even design or implementation models, leaving the prototype as something that can be destroyed. This causes several major problems:

1. Since prototypes are destined to be destroyed, no importance is attached to their design and construction. As a result, in many cases, not enough resources are allocated to produce a prototype of high quality.
2. Precisely because of this lack of quality and precision in prototype development, definitions may be established which are either ambiguous or incomplete, covering only part of the development.
3. If the prototype is destroyed, it is difficult to trace the origins of customers’ and users’ requirements for future changes or problems. Here, one recommended procedure guided by good practice is that of the CMMI model (Aggarwal et al., 2014; Gupta, 2011).

In view of the above concerns, a need has arisen to investigate whether mechanisms exist which allow us to conceive prototypes as something that is “usable” in subsequent phases and consequently supportive to the development of a tool. If a development team could automatically (or semi-automatically) use all the knowledge that a prototype provides, more effort should be devoted to it, with the guarantee that it is an investment for the future. Improving prototypes would allow us to improve communication with users and increase the quality of requirements models.

This work addresses this premise and aims to analyze the solutions that have been proposed in the literature for using prototypes efficiently and effectively. The objective is to find out how analysis arti-

23 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:
www.igi-global.com/chapter/getting-more-out-of-software-prototypes/343650

Related Content

Analyzing Consumer Impulse Purchasing Behaviour Using Observational Data

Yuliia Kyrdoda, George Baltasand A.Malek Hammami (2018). *International Journal of Food and Beverage Manufacturing and Business Models* (pp. 16-28).

www.irma-international.org/article/analyzing-consumer-impulse-purchasing-behaviour-using-observational-data/210635

ArghCompetence Recognition and Self-Concept of Employees on Motivation and its Impact

Irwan Usman, Haris Maupa, Sitti Haeraniand Muhammad Idrus Taba (2020). *International Journal of Applied Management Theory and Research* (pp. 48-60).

www.irma-international.org/article/arghcompetence-recognition-and-self-concept-of-employees-on-motivation-and-its-impact/244219

Relationship Between Empowerment and Organizational Commitment: An Empirical Study of IT Industry

Bhanu Priyaand Vandana Singh (2018). *Management Techniques for a Diverse and Cross-Cultural Workforce* (pp. 172-185).

www.irma-international.org/chapter/relationship-between-empowerment-and-organizational-commitment/198230

Prediction of the Disappearance of Companies From the Market in Bogotá, Colombia Using Machine Learning

William Stive Fajardo-Moreno, Rubén Dario Acosta Velásquez, Ivan Dario Castaño Pérezand Leonardo Espinosa-Leal (2021). *Handbook of Research on Management Techniques and Sustainability Strategies for Handling Disruptive Situations in Corporate Settings* (pp. 227-246).

www.irma-international.org/chapter/prediction-of-the-disappearance-of-companies-from-the-market-in-bogot-colombia-using-machine-learning/285878

Evolution of Supply Chain Collaboration: Implications for the Role of Knowledge

Michael J. Gravierand M. Theodore Farris (2014). *Management Science, Logistics, and Operations Research* (pp. 333-374).

www.irma-international.org/chapter/evolution-of-supply-chain-collaboration/97006