Chapter 6
Benefits and Challenges in the Use of Case Studies for Security Requirements Engineering Methods

Nancy R. Mead
Carnegie Mellon University, USA

ABSTRACT

The premise of this paper is that pilot case studies in security requirements engineering provide both benefits and challenges to the underlying research, education, and technology transition effort. Over the past four years we have worked with seven development groups in five organizations in the process of refining and transitioning the Security Quality Requirements Engineering (SQUARE) and SQUARE-Lite methods into practice. These experiences have provided the opportunity to step back and assess the use of pilots in conjunction with student projects to support method refinement and technology transition. Although SQUARE and SQUARE-Lite are concerned with security requirements, the benefits and challenges that have been observed would apply to many security research and technology transition efforts. We itemize and justify these benefits and challenges and discuss their practical relevance and application to ensuring adequate information assurance protection.

INTRODUCTION

In this paper we discuss the role of pilot case studies in security requirements engineering and their impact on method refinement, student projects, and technology transition. We start by providing some general background on the importance of requirements engineering and some specifics on the problems encountered in security requirements engineering. We then introduce the SQUARE and SQUARE-Lite methods, which were the research models used in our case studies. Next we discuss

DOI: 10.4018/978-1-4666-1580-9.ch006

the case studies, their organizations, and general case study results. We provide detailed results for one of the case studies. We go on to illustrate

- The use of the case studies in refining the SQUARE and SQUARE-Lite methods. We found that the case studies allowed us to identify issues in both methods, resulting in revision of the methods.
- The use of research projects as the basis for student projects. Benefits included the opportunity to work on real client projects, and to provide feedback to both the clients and the research project. Challenges included the need to limit the case studies to a single semester, and to help the students deal with the uncertainties of working on new research.
- The benefits and challenges associated with technology transition of new methods such as SQUARE. Benefits included the opportunity to impact organizational software processes and to provide projects with new insights into security requirements engineering. Challenges included the difficulty of getting busy staff members to work with us and the uphill battle to effect change in security requirements engineering practice, in the absence of larger organizational change.

Although none of these is unique, we seldom look at these three aspects in a unified way, and we almost never discuss the difficulties, since we are generally motivated to discuss our successes.

BACKGROUND

It comes as no surprise that requirements engineering is critical to the success of any major development project (Mead, 2008b). Some studies have shown that requirements engineering defects cost 10 to 200 times as much to correct once fielded than if they were detected during requirements development (Boehm & Papaccio, 1988; McConnell, 2001). Other studies have shown that reworking requirements, design, and code defects on most software development projects costs 40 to 50% of total project effort (Jones, 1986), and the percentage of defects originating during requirements engineering is estimated at more than 50%. The total percentage of project budget due to requirements defects is 25 to 40% (Wiegers, 2003).

A prior study found that the return on investment when security analysis and secure engineering practices are introduced early in the development cycle ranges from 12 to 21%, with the highest rate of return occurring when the analysis is performed during application design (Berinato, 2002). The National Institute of Standards and Technology (NIST) reports that software that is faulty in security and reliability costs the economy $59.5 billion annually in breakdowns and repairs (National Institute of Standards and Technology, 2002). The costs of poor security requirements show that even a small improvement in this area would provide a high value. By the time that an application is fielded and in its operational environment, it is very difficult and expensive to significantly improve its security.

Requirements problems are among the top causes (Charette, 2005) of why

- Projects are significantly over budget
- Projects are past schedule
- Projects have significantly reduced scope or are cancelled
- Development teams deliver poor-quality applications
- Products are not significantly used once delivered

Security requirements are often identified during the system life cycle. However, the require-