

Extreme Programming for Mobile Applications

E

Pankaj Kamthan

Concordia University, Canada

INTRODUCTION

The liberty, expediency, and flexibility that come with mobile access have led to proliferation of mobile applications. At the same time, these applications face constant challenges posed by new implementation languages, variations in user agents, and demands for new services from user classes of different cultural backgrounds, age groups, and capabilities.

To address that, we require a methodical approach towards the development lifecycle and maintenance of mobile applications that can adequately respond to this constantly changing environment. In other words, it needs to be *agile* (Highsmith, 2002). In this article, we propose the use of an agile methodology, Extreme Programming (XP) (Beck & Andres, 2005), for a systematic development of mobile applications.

The organization of the article is as follows. We first outline the background necessary for the discussion that follows and state our position. This is followed by a discussion of the applicability and feasibility of XP practices as they pertain to mobile applications. Then the limitations of XP towards mobile applications, particularly those that are developed in an open source setting, are highlighted, and suggestions for improvement are presented. Next, challenges and directions for future research are outlined. Finally, concluding remarks are given.

BACKGROUND

In recent years, ongoing efforts towards affordability of mobile devices by public-at-large and increasing contact points (service providers) have opened new vistas in the arena of mobile applications. This has also resulted in increased expectations, including sophisticated services, from mobile users. As a consequence, mobile applications continue to become increasingly large and complex. This growth, however, needs to be carefully controlled and sustained. For that, it is important that the lessons learned from the successes and failures (Nguyen, Johnson, & Hackett, 2003) in the evolution of Web applications not be ignored. Specifically, a systematic approach for creating mobile applications is desirable. We call this *Mobile Web Engineering*, inspired by traditional software engineering and Web engineering (Ginige & Murugesan, 2001).

The focus in the literature (Hjelm, 2000), however, has primarily been on implementation languages rather than the *process*. In Salmre (2005), a systematic approach to developing mobile applications is advocated, but the discussion is within the technology-specific context of Microsoft .Net Framework and Visual Basic. It is unclear how these can scale to the changing technological environment. One of the purposes of this article is to fill this gap.

We adopt the most broadly used and well-tested agile methodology, namely XP, for the development of mobile applications. XP is a test-driven “lightweight” methodology designed for small teams which emphasizes customer satisfaction and promotes teamwork. XP was created to tackle uncertainties in development environment, and in doing so, put more emphasis on the social (people) component (engineer, customer, and end user). The XP practices are set up to mitigate project risks (dynamically changing requirements, new system due by a specific timeline, and so on) and increase the likelihood of success. The use of XP has been suggested for a “rapid application development” of Web applications (Wallace, Raggett, & Aufgang, 2002; Maurer & Martel, 2002).

It is not the purpose of this article to evaluate the merit of XP on its own or with respect to other agile methodologies; such assessments have been carried out elsewhere (Turk, France, & Rumpe, 2002; Mnkanla & Dwolatzky, 2004).

ENGINEERING MOBILE APPLICATIONS USING EXTREME PROGRAMMING

In this section, we discuss in detail how the practices put forth by XP manifest themselves in the development of mobile applications (see Table 1). The 12 XP practices are: *The Planning Game*, *Small Releases*, *Metaphor Guide*, *Simple Design*, *Testing*, *Refactoring*, *Pair Programming*, *Collective Ownership*, *Continuous Integration*, *40-Hour Week*, *On-Site Customer*, and *Coding Standards*.

We note that some of these practices such as *Testing*, *Refactoring*, or *Pair Programming* are not native to XP and were discovered in other contexts previously. In this sense, by aggregating them in a coherent manner, XP bases itself on “best practices.” These practices are also not necessarily mutually exclusive, and we point out the relationships among them where necessary. We also draw attention to the obstacles

Table 1. XP practices corresponding to process workflows in a mobile application

Process Workflow	XP Practices
Planning	40-Hour Week, The Planning Game (Project Velocity)
Analysis (Domain Modeling, Requirements)	On-Site Customer, The Planning Game (User Stories)
Design	Metaphor Guide (Natural Naming), Simple Design, Refactoring
Implementation	Collective Ownership, On-Site Customer, Metaphor Guide, Coding Standards, Pair Programming, Continuous Integration
Verification and Validation	On-Site Customer, Testing (Unit Tests, Acceptance Tests)
Delivery	Small Releases

in the realization of these practices that pose challenges to the deployment of XP for mobile applications.

The Planning Game

The purpose of *The Planning Game* is to determine the scope of the project and future releases by combining business priorities and technical estimates. For that, it solicits input from the “customer” to define the business value of desired features and uses cost estimates provided by the programmers. This input comes in form of *user stories* (Alexander & Maiden, 2004). A user story is a user experience informally expressed in a few lines with a mobile application such as navigating or using a search engine. The estimation is limited to the assessment of *project velocity*, a tangible metric that determines the pace at which the team can produce deliverables. The plan is prone to modifications based on the current reality.

Small Releases

The idea behind *Small Releases* is to have a simple system (an evolutionary prototype) into production early, and then via short cycles, iteratively and/or incrementally, reach the final system. To have a concrete proof-of-concept up and running can be used to solicit feedback for future versions and can help convince customers and managers of the viability of the project. This is useful for mobile applications that are highly interactive. However, there is cost associated with prototypes and therefore their number should be kept under control.

Metaphor Guide

The use of metaphors (Boyd, 1999) is prevalent in all aspects of software development. A *Metaphor Guide* is an effort to streamline and standardize efforts for naming software objects and is available for team-wide use. Natural naming

(Keller, 1990) is a technique initially used in source code contexts that encourages the use of names that consist of one or more full words of the natural language for program elements in preference to acronyms or abbreviations. Indeed, natural naming strengthens the link between the underlying conceptual entity and its given name. For example, *MobileProfile* is a combination of two real-world metaphors placed into a natural naming scheme. The two main concerns in naming are: (1) length due to the constrained interfaces on mobile devices, and (2) user familiarity, as user background is often non-technical. For example, it is preferable to use *EnterSearchWords* as an indicator inside the form interface (to save space) rather than *RegularExpressionForQueryString* outside the form, although the latter may be a more accurate description.

Simple Design

The motivation behind a *Simple Design* is that in XP’s view, requirements are *not* complete when the design commences. This is in line with the reality of mobile applications, which have to respond to the market pressures and the competition that are beyond their control, or other unavoidable circumstances such as variations in implementation technology. Therefore, the design is minimal based on *current* (not future) requirements. It aims for simplicity, and to ensure “good” design its quality (specifically, structural complexity) is improved by frequent revisitations, that is, *Refactoring*.

Testing

There is a strong emphasis in XP on validation and verification of the software at all times. By being test driven, there is transition from one phase to another only if the tests succeed. The tests range from unit tests (using tools such as HTMLUnit, HTTPUnit, XMLUnit, XSLTUnit, and JUnit) written by programmers to acceptance tests involving customers (to satisfy customer requirements). There are variations in user

4 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/extreme-programming-mobile-applications/17089

Related Content

Trust Management and Modeling Techniques in Wireless Communications

Revathi Venkataraman, M. Pushpalatha and T. Rama Rao (2014). *Security, Privacy, Trust, and Resource Management in Mobile and Wireless Communications* (pp. 278-294).

www.irma-international.org/chapter/trust-management-and-modeling-techniques-in-wireless-communications/86309

Older Adults with AMD as Co-Designers of an Assistive Mobile Application

Lilit Hakobyan, Jo Lumsden and Dymphna O'Sullivan (2014). *International Journal of Mobile Human Computer Interaction* (pp. 54-70).

www.irma-international.org/article/older-adults-with-amd-as-co-designers-of-an-assistive-mobile-application/107990

Awareness of Mobile Device Security: A Survey of User's Attitudes

Nathan Clarke, Jane Symes, Hataichanok Saevanee and Steve Furnell (2016). *International Journal of Mobile Computing and Multimedia Communications* (pp. 15-31).

www.irma-international.org/article/awareness-of-mobile-device-security/148259

Using Mobile Technologies as Research Tools: Pragmatics, Possibilities and Problems

Ria Hanewald (2011). *Mobile Technologies and Handheld Devices for Ubiquitous Learning: Research and Pedagogy* (pp. 99-119).

www.irma-international.org/chapter/using-mobile-technologies-research-tools/46561

Mobile Payment and Its Social Impact

Liguo Yuan and Liping Sun (2018). *Mobile Applications and Solutions for Social Inclusion* (pp. 157-182).

www.irma-international.org/chapter/mobile-payment-and-its-social-impact/204714