A Comparison of Use Cases and User Stories

Pankaj Kamthan
Concordia University, Canada

INTRODUCTION

The discipline of software engineering advocates a systematic and disciplined approach towards the development and evolution of software systems. The ecosystem of software engineering is constantly changing, and influences the practice of software engineering. In the past decade, there have been a number of notable strategic changes in industrial software engineering, including the adoption of agile methodologies (Highsmith, 2009).

There are a number of concerns in agile software development, including requirements engineering (Leffingwell, 2011; Sillitti & Succi, 2005; Zhu, 2009). There are different types of agile requirement, of which currently the most common forms are *use cases* (Jacobson et al., 1992) and *user stories* (Cohn, 2004). Even though use cases and user stories have different origins, both in space and in time, the two are not entirely unrelated. The purpose of this article is to place use cases and user stories in context of each other.

The need for a comparison between concepts arises naturally in many disciplines, and agile requirements engineering is no exception. Indeed, it is suggested by theories of learning, such as constructivism (Piaget, 1952), that upon initial exposure to a new concept C, a comparison between C and other closely related and relatively more established concepts C' is inevitable. This comparison is a necessary prerequisite for creating an understanding of C through assimilation and accommodation. In this article, C is user story and C' is use case.

The rest of the article is organized as follows. First, a brief background on agile methodologies and previous work on comparing use cases and user stories is presented. This is followed by introduction of a framework for systematically comparing use cases and user stories. Next, directions for future research are highlighted. Finally, concluding remarks are given.

BACKGROUND

In the 1990s, a number of limitations of rigidity in approaches for development of certain types of software systems were realized. The drive to cope with these limitations led to the inception of agility.

The *Agile Manifesto* constitutes the basis for a number of agile methodologies, including Agile Experience Design (AXD), Crystal Clear, Extreme Programming (XP), OpenUP, Scrum, and User-Centered Agile Process (UCAP). In certain cases, such as the Discipline Agile Delivery (DAD) process framework (Ambler & Lines, 2012), elements of multiple agile methodologies have been included.

In general, in software engineering, significant emphasis is placed on requirements engineering in the light of its impact on later phases of software development (Wiegers, 2003). In particular, in agile requirements engineering, the attention is on the problem so as to devise a desirable and viable solution, namely the software system, which can satisfy the stakeholders.

The agile methodology underlying an agile project determines the type of agile requirement to be adopted. For example, in Crystal Clear and OpenUP, an agile requirement can take the form of a use case (Jacobson et al., 1992) and in AXD, XP, Scrum, and UCAP, an agile requirement can take the form of a user story (Cohn, 2004).

Related Work

The need for understanding the differences between use cases and user stories has been, directly or indirectly, expressed in several places in literature. In the following, previous efforts that compare use cases and user stories are analyzed briefly and chronologically.

The notion of simplicity has been a criterion for comparison between use cases and user stories. In one of the earliest work on user stories (Beck, 2000), it

has been pointed out that user stories are "simplified use cases." However, the meaning of 'simplification' and exactly what is "simplified" has not been given.

The structure of the means used for a description has often been a criterion for comparison between use cases and user stories. Indeed, it has been stated that use cases are "structured" while user stories (in XP) are "unstructured" (Cohn & Paul, 2001), that "use cases are a structured representation of a user story" (Decker et al., 2006), and that "[use cases] are expressed using a constrained (semi-formal) syntax" and that "[user stories] are expressed using natural language prose" (Alexander & Maiden, 2004). However, there are a number of ways of expressing a use case (Cockburn, 2001), and not all of them need to be "structured" or follow a "constrained (semi-formal)" syntax.

The issue of a software requirement treading into software design has been a criterion for comparison between use cases and user stories. For example, it has been asserted that "use cases are more prone to including details of the user interface" (Cohn, 2004) and that "it is usual for use cases to include user interface details" (Monochristou & Vlachopoulou, 2007). However, there are guidelines (Cockburn, 2001; Wiegers, 2003) that explicitly suggest against the inclusion of any user interface specifics in use cases.

Finally, based on a controlled experiment, it has been concluded that use cases can be useful as a complement to user stories (Gallardo-Valencia, Olivera, & Sim, 2007).

The issues underlying related work can be summarized as follows. The comparison between use cases and user stories is treated as a secondary, not a primary, concern; the comparison is not systematic; and the instrument for comparison has a singular perspective. This article is motivated, in part, by the need to address these issues. To do that, a systematic approach for comparison that rests on the foundations of software engineering is required.

A FRAMEWORK FOR COMPARING USE CASES AND USER STORIES

In this section, use cases and user stories are compared systematically. The purpose of comparison is to facilitate understanding and appropriate use of each.

Table 1. A framework for comparing use cases and user stories

Discipline	Viewpoint	Criterion
Software Engineering	Project	● Estimate ● Schedule
	Process	● Methodology ● Development
	People	● Users ● Value
	Product	● Scope ● Description
	Resource	● Purpose ● Maturity

The comparison is based on a framework, as given in Table 1, consisting of a set of meta-criteria that are decomposed into a set of criteria. The meta-criteria consist of certain viewpoints considered relevant to software engineering by the IEEE Software and Systems Engineering Standards Committee. The criteria are selected in a manner that they are relevant to software engineering, as well as both to use cases and to user stories. The criteria are considered equally significant.

The rest of the section provides details of comparison between use cases and user stories, organized by viewpoints and criteria, highlighting the similarities and differences between them in the process.

Use Cases vs. User Stories: Project-Viewpoint

A comparison of use cases and user stories from a project viewpoint can be based on the following criteria: estimate and schedule.

Criterion: Estimate

Use Cases: In light of their availability early in a software development process, use cases have been recommended as a basis for estimation (Karner, 1993; Mohagheghi, Anda, & Conradi, 2005). The purpose of estimation is to schedule the entire software project. This approach requires the presence of all use cases upfront, and the fact that they all are expressed in a certain manner, for calculating the estimate. However, a use case is not intrinsically related to estimation. Furthermore, software development methodologies that

Related Content

Accessibility Solutions for Visually Impaired Persons: A Digital Platform Conceptualization
Rita Oliveira, Alcina Prata, José Carlos Miranda, Jorge Ferraz de Abreuand Ana Margarida Almeida (2021). *Handbook of Research on Multidisciplinary Approaches to Entrepreneurship, Innovation, and ICTs (pp. 331-348).*
www.irma-international.org/chapter/accessibility-solutions-for-visually-impaired-persons/260564

Big Data Summarization Using Novel Clustering Algorithm and Semantic Feature Approach
Shilpa G. Kolteand Jagdish W. Bakal (2017). *International Journal of Rough Sets and Data Analysis (pp. 108-117).*
www.irma-international.org/article/big-data-summarization-using-novel-clustering-algorithm-and-semantic-feature-approach/182295

Statistics on Image Engineering Literatures
Yu-Jin Zhang (2015). *Encyclopedia of Information Science and Technology, Third Edition (pp. 6030-6040).*
www.irma-international.org/chapter/statistics-on-image-engineering-literatures/113059

Sociotechnical Change Perspective for Enterprise Resource Planning System Implementation
Jessy Nair, D. Bhanu Sree Reddyand Anand A. Samuel (2015). *Encyclopedia of Information Science and Technology, Third Edition (pp. 923-933).*
www.irma-international.org/chapter/sociotechnical-change-perspective-for-enterprise-resource-planning-system-implementation/112485

An Efficient and Simple Algorithm for Matrix Inversion
Ahmad Farooqand Khan Hamid (2012). *Knowledge and Technology Adoption, Diffusion, and Transfer: International Perspectives (pp. 21-28).*
www.irma-international.org/chapter/efficient-simple-algorithm-matrix-inversion/66932