A Characterization of Negative User Stories

Pankaj Kamthan
Concordia University
Montreal, Canada
kamthan@cse.concordia.ca

Nazlie Shahmir
WestJet Airlines Limited
Calgary, Canada
nshahmir@westjet.com

***Abstract*—In the context of an agile project, negative interactions are addressed by equipping the 'conventional' positive user story engineering process with a number of conceptual models, including those for negative user story and negative role. The challenges inherent in eliciting negative uses, negative roles, and negative user stories are highlighted. The cost of engineering negative user stories is analyzed. The relationships among positive and negative user stories are considered. For illustration, a detailed example is presented.**

Keywords-agile methodologies; conceptual modeling; requirements engineering; risk management; security engineering; socio-technical distributed systems

I. INTRODUCTION

The E-Type software systems are a large class of systems that reflect human processes or the real world [1]. In recent years, the agile methodologies are being used increasingly in the industry for the development of E-Type software systems aimed for general public consumption, such as distributed software systems, in general, and interactive Web Applications, in particular [2].

If an agile project team anticipates a positive *user experience* (UX), then potential negative uses and their impact need to be *forethought* and need to be essential concerns throughout the agile process. The purpose of this paper is to serve as a starting point towards such an endeavor by proposing a 'network' of conceptual models that provide necessary abstraction and input to different steps in modeling negative users and potential negative uses to support the positive *user story engineering* process.

The rest of the paper is organized as follows. In Section II, background is provided and related work is highlighted. The elements of negative user story engineering are introduced in Section III. In Section IV, directions for future research are outlined. Finally, in Section V, concluding remarks are given.

II. BACKGROUND AND RELATED WORK

The negative uses of software systems predate the Internet, but have been catalyzed by the broad acceptance by general public and use of the Web for activities beyond for which it was envisioned originally. The negative uses can be different kinds. Traditionally, the notion of negative use has been related intimately to *privacy* (about preventing psychological harm), *security* (about preventing possessional harm), and/or *safety* (about preventing physiological harm). The impact of such negative uses ranges from *innocuous* (annoying and/or distracting) to *nocuous* (cognitively, emotionally, financially, legally and/or socially damaging), temporarily or permanently. The negative uses seriously undermine the otherwise increasingly important role played by software for the benefit of society.

In agile methodologies, software requirements are usually expressed as either *user stories* or lightweight *use cases*. In the rest of the paper, 'user story' and 'positive user story' are considered synonymous and used interchangeably.

In the past decade or so, addressing negative use during (agile) requirements engineering has garnered much interest. For example, the notion of *abuse case* [3] and the concept of (and graphical modeling notation for) *misuse case* [4] have been proposed and have been used for the elicitation of security requirements. XP practices have been extended to support the creation of security-related user stories that are informed by a risk assessment of *abuser stories* [5]. However, the attention on the abuser and the form for expressing abuser stories is inadequate. There is preliminary work on modeling negative user stories [6], which, in part, forms the motivation for this paper.

III. ELEMENTS OF NEGATIVE USER STORY ENGINEERING

A. Conceptual Models

The compendium of conceptual models that follows is a requisite for a comprehensive understanding and systematic engineering of negative user stories. These models, of which some appear in an earlier work [6], are intrinsically interrelated and are informed by international standards.

1) Context of Use Model

This model provides an understanding of the technical as well as non-technical *environment factors* under which a user uses the software system. The technical factors include network connection type, device type, and operating system type. The non-technical factors include mental and physical ability of the user. For example, the ISO 9241-210 Standard can be a source for such a model.

2) Positive User Story Model

This model provides an understanding of the notion of positive user story, and highlights the elements necessary for expressing a positive user story properly [7]. For uniformity, a positive user story statement could be structured in controlled natural language text as: A role can goal to value. The goal must be explicit; the value may be implicit or explicit.

DOI reference number: 10.18293/SEKE2016-106

3) Positive Role Model

This model provides an understanding of the characteristics and behavior of a typical positive user (playing a particular role) [6]. For example, a *persona* (an archetypical user of a software system) can be such a model [8]. It helps create empathy among requirements engineers towards positive users.

4) Negative Use Model

This model provides an understanding of the types of negative uses that a software system could be subjected to, their probabilities of occurrence, and their consequences, if realized [6]. It helps create awareness of negative uses among requirements engineers, and provides the knowledge (including terminology) necessary for expressing a negative user story properly.

5) Negative User Story Model

This model provides an understanding of the notion of negative user story, and highlights the elements necessary for expressing a negative user story properly [6]. For uniformity, a negative user story statement could be structured in controlled natural language text as: A negative role wants to negative goal to negative value. A negative user story is not designed or implemented. It therefore has no acceptance criteria or estimate, but is associated with a risk assessment.

6) Risk Assessment Model

This model provides an understanding of risk assessment to be able to make informed decisions about negative user stories. According to the ISO Guide 73, a *risk source* is the element that has the intrinsic potential to give rise to risk. For example, unprotected credit card information of a customer is a risk source. A *risk* is the combination of the probability in the interval (0, 1) of a *threat* (a circumstance with the potential to produce loss) and its *consequence* (the loss that will be incurred if the corresponding threat is realized). A *risk exposure* (RE) is the potential loss incurred by a risk.

RE is a function of the likelihood of the threat and the impact of its consequence. Using these as the two orthogonal dimensions, RE can be given qualitatively by a *risk matrix*, as shown in Fig. 1. In this case, the possible RE levels are N (Negligible), L (Low), M (Medium), H (High), and E (Extreme). RE can serve as a basis for *risk assessment*. For example, RE level of "M", "H", or "E" could be seen as significant, whereas RE level of "N" or "L" could be seen as insignificant.

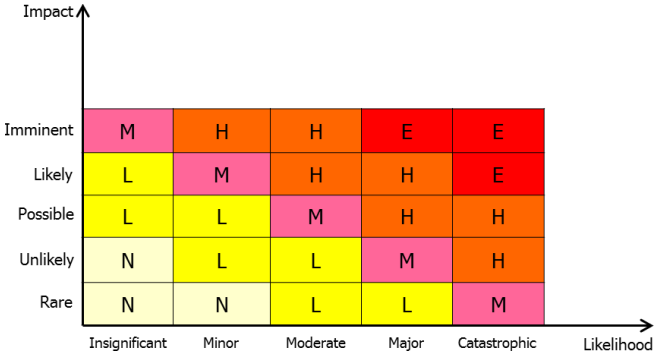

Figure 1. A risk matrix.

7) Negative Role Model

This model provides an understanding of the characteristics and behavior of a typical negative user (playing a particular role) [6]. For example, a *negative persona* (an archetypical negative user of a software system) can be such a model [8]. The negative roles can be of different kinds, including *Fraudster*, *Imposter*, *Malicious Hacker* (*Black Hat Hacker*), *Phisher*, *Prankster*, *Spammer*, and *Vandal*. There is currently no standard classification of negative roles. The actions of a negative role are unethical, but may or may not be unlawful. For example, the actions of a black hat hacker are normally unlawful, but that of a prankster are not.

B. Challenges in Eliciting Negative Uses, Negative User Stories, and Negative Roles

The development of a software system can be viewed as acquisition of knowledge throughout the (agile) process. In a model of knowledge (or lack thereof, that is, ignorance) of a person, there are a number of different states in the increasing order of ignorance [9]:

- **Zeroth-Order Ignorance (0OI)—Lack of Ignorance:** "A person knows something." For example, a software engineer knows about a vulnerability in the application programming interface (API) of a programming language used in the implementation.
- **First-Order Ignorance (1OI)—Lack of Knowledge:** "A person knows that he or she does not know something." For example, a software engineer knows that he or she does not know the motives of a negative role or the time a negative role may interact with the software system, or the irreparable financial loss or the emotional impact that can incur on the customers due to negative use.
- **Second-Order Ignorance (2OI)—Lack of Awareness:** "A person does not know that he or she does not know something." For example, a software engineer does not know that he or she does not know about the criticality of certain security-related defects missed during testing.

To 'think negatively' and be able to anticipate all possible negative uses of a software system can be difficult for software engineers, even if they are trained in critical thinking.

The candidate negative roles and the candidate negative user stories could, as usual, be identified, classified, and prioritized, using *ideation techniques* (such as brainstorming and mind mapping). However, certain *ethnographic techniques* (such as interviews and surveys), which have proven to be useful for positive roles and positive user stories, cannot be applied effectively for negative roles and the candidate negative user stories.

It may be difficult to prevent negative uses entirely [2]. Indeed, with the evolution of the Social Web and increasing use of distributed software systems, the inception of new types of negative uses (such as social engineering) is inevitable and the number of negative uses is unlikely to decrease. Even if a negative role's motives or ways may be unfamiliar (due to 1OI or 2OI), it can be expected that he or she exploits vulnerabilities underlying familiar aspects of

software, such as boundary conditions, intersystem communication, and system assumptions. For example, in the context of a Web client-server environment, if HTTP cookies are relied on exclusively for user identification and if it is assumed that the Web client never modifies its HTTP cookies before they are sent back to the requesting Web server, then a negative role could cause problems by taking control of the session and modifying the HTTP cookies.

C. A User Story Engineering Process for Negative Uses and its Implications

The 'conventional' positive user story engineering process needs to be *extended* to accommodate negative user stories, as shown in Fig. 2. The conceptual models introduced in the previous section are an input to this process.

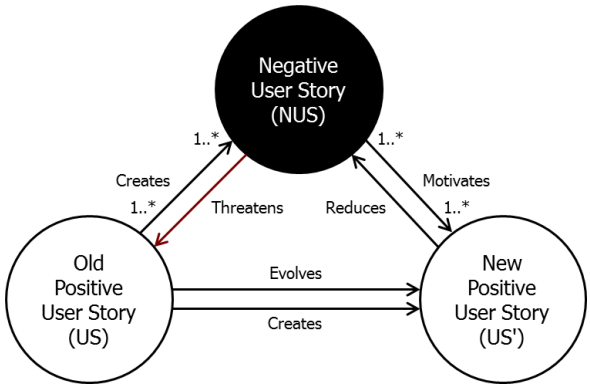

Figure 2. A negative user story influences a positive user story.

The dynamics of the process in Fig. 2 can be explained as follows. Let there be a positive user story, US. The existence of US creates the potential for a negative user story, NUS, which, in turn, threatens a successful realization of US, and therefore prompts a change to the US. If a risk assessment concludes a significant RE level of NUS, then there are two options for *reducing* the risk posed by NUS: (1) evolve old positive user story, US to US', and/or (2) create new positive user story, US'. For example, (1) could involve refining the text of US and/or splitting US (into two or more user stories).

D. Economics of the User Story Engineering Process for Negative Uses

The addition of a negative user story in the 'conventional' positive user story engineering process entails cost that, for the sake of feasibility, need be weighed against the perceived benefits. There are different kinds of costs associated during the positive user story engineering process that involves a negative user story:

1. There is cost, say C_1, if the negative goal and negative value of the negative user story are satisfied. This cost could, for example, be measured in terms of customer dissatisfaction.
2. There is cost, say C_2, of risk assessment.
3. There is cost, say C_3, of evolving old positive user stories or eliciting new positive user stories, as the case may be.

The probability of incurring C_1 belongs to a continuous set, whereas the probabilities of incurring C_2 or C_3 belong to a discrete set.

There are three cost scenarios:

Cost Scenario 1: If there is no risk assessment, then there is no need for evolving old positive user stories or eliciting new positive user stories. In this case, there is no C_2 or C_3, but there is still C_1. In this case, it is assumed that $C_1 < C_2 + C_3$.

Cost Scenario 2: If a risk assessment suggests an insignificant RE level, then there is no need for evolving old positive user stories or eliciting new positive user stories. In this case, there is no C_3, but there is still C_1 and C_2. In this case, it is assumed that $C_1 + C_2 < C_3$.

Cost Scenario 3: If a risk assessment suggests a significant RE level, then there is a need for evolving old positive user stories or eliciting new positive user stories. In this case, there is C_2 and C_3, and, hopefully, $C_2 + C_3 < C_1$.

E. Example

Let there be an electronic shopping system being developed as a Web Application using one of the agile methodologies that has inherent support for user stories. This shopping system may typically have a shopping cart system, a payment system, and a customer identity management system.

The existence of a positive user story, US-0, creates the potential for a negative user story, NUS.

US-0. A customer can make a payment using a credit card to purchase a shopping item.

NUS-0. A fraudster wants to steal credit card information to make unauthentic charges.

The initiative to reduce the risk posed by NUS to US-0 motivates (at least) the following interrelated positive user stories: US-1, which is an evolution of US-0, and US-2 and US-3, which are new creations (assuming they did not exist).

US-1. A customer can receive an e-mail message asking him or her to confirm a purchase using a credit card to shop assuredly.

US-2. A customer can create an account on the shopping system to shop serenely.

US-3. A customer can receive an e-mail address as part of his or her account on the shopping system to send messages to and receive messages from the shopping system.

It is customary for a positive user story to be associated with *acceptance criteria*. The acceptance criteria for US-1 needs to consider NUS-0, and, in doing so, must contain appropriate tests to ensure that the purchase is legitimate before processing the payment. For example, one test could check the customer's e-mail address is indeed in his or her profile and another test could check the contents of customer's response to the e-mail message.

F. Positive and Negative User Story Relationships

The realization of a positive user story in the software system creates the potential for one or more negative user stories. For example, existence of US-1 creates the potential for a negative user story involving clone phishing (specifically, e-mail spoofing).

The converse is also holds, that is, a negative user story is always associated with one or more positive user stories. For example, if there is no provision for making a payment using a credit card, then there is no need for US-0, and, in turn, no potential for credit card fraud in the given context and, therefore, no possibility of NUS-0.

G. Impact of Negative User Stories on the Positive User Story Relationships

The software requirements, in general, and positive user stories, in particular, can be interrelated in many different ways, such as *Causal*, *Essential* (Is-Dependent-On, Is-Constrained-By), *Implementational* (Conflicts-With, Is-Cost-Related-To, Is-Similar-To, Is-Value-Related-To), *Spatial* (Is-Aggregated-With, Is-Generalized-To, Is-Refined-By), and *Temporal* (Is-Sequential-To, Is-Interleaving-With, Is-Synchronized-With). For example, US-2 is an aggregate of a number of positive user stories, including US-3 and US-4.

US-4. A customer can supply his or her postal address as part of his or her account on the shopping system to receive shopping items.

For software engineers, an in-depth understanding of interrelationships among positive user stories is necessary for a number of reasons, such as prioritizing, scheduling, and release planning the positive user stories properly, testing the positive user stories adequately, and modifying non-independent positive user stories relatively easily.

The existence of a negative user story can influence ways in which positive user stories are interrelated. If a negative user story leads to a change of a positive user story (say, US), then, evidently, the other positive user stories that depend in some way on US can be affected and may need to be changed, too. These changes may not be isolated and can, in fact, *propagate* due to the presence of certain properties (such as symmetry and/or transitivity) of relationships. For example, if there is a set of positive user stories related by the Is-Synchronized-With relationship, then a negative user story that affects one positive user story in that set will affect all positive user stories in that set, as the Is-Synchronized-With relationship is symmetric as well as transitive.

IV. Directions for Future Research

A. Other Negative Uses

In recent years, agile methodologies are being applied for the development of airline reservation systems and healthcare information systems [10]. These systems need to be concerned with privacy, safety, and security, and, at the same time, be able to provide a positive UX.

For example, for an airline reservation system, a realization of the following negative user story is (a case of eavesdropping and therefore) a violation of privacy:

NUS-1. A prankster wants to intercept an HTTP cookie to be able to monitor people's travelling habits.

For another example, for a healthcare information system, a realization of the following negative user story is a violation of privacy, safety, and security:

NUS-2. An identity thief wants to steal the medical records of patients to coerce.

Therefore, exploring homogeneous and heterogeneous combinations of violations of privacy, safety, and security, with due consideration for accessibility and usability, is of research interest.

B. Empirical Studies

Traditionally, most agile methodologies do not have native support for extensive conceptual modeling. Therefore, highlighting the human and social challenges inherent in the constructions of the conceptual models is of research interest. Finally, for assessing the feasibility of a deployment of the positive user story engineering process extended by negative user stories, it can be useful to conduct empirical studies in organizations with appropriate agile process maturity levels.

V. Conclusion

It is crucial for the organization to cultivate a culture for *proactively* and *cost-effectively* identifying, understanding, and (hopefully) preventing, negative uses of the products and services it provides, so as to sustain confidence of its customers and other stakeholders, to manage its requirements debt, to retain its share and reputation in the market, and to be perceived as socially responsible.

In the context of an agile project, such a commitment requires adequate preparation as early as possible, that is, during conceptual modeling for understanding the problem and positive user story engineering. In particular, if a negative user story with a significant RE level is identified, then, as this paper has attempted to show, it is incumbent upon the development team to take preventative measures.

References

[1] K. Duran, G. Burns and P. Snell, Lehman's Laws in Agile and Non-Agile Projects. The Twentieth Working Conference on Reverse Engineering (WCRE 2013), Koblenz, Germany, October 14-17, 2013.

[2] W. Kim, O.-R. Jeong, C. Kim and J. So, The Dark Side of the Internet: Attacks, Costs and Responses. Information Systems, 36(3): 675-705, 2011.

[3] J. McDermott and C. Fox, Using Abuse Case Models for Security Requirements Analysis. The Fifteenth Annual Computer Security Applications Conference (ACSAC 1999), Scottsdale, USA, December 6-10, 1999.

[4] G. Sindre and A. L. Opdahl, Eliciting Security Requirements by Misuse Cases. Requirements Engineering, 10(1): 34-44, 2005.

[5] J. Peeters, Agile Security Requirements Engineering. The Symposium on Requirements Engineering for Information Security (SREIS 2005), Paris, France, August 29, 2005.

[6] P. Kamthan and N. Shahmir, Modeling Negative User Stories is Risky Business. The Seventeenth IEEE International Symposium on High Assurance Systems Engineering (HASE 2016), Orlando, USA, January 7-9, 2016.

[7] M. Cohn, User Stories Applied: For Agile Software Development. Addison-Wesley, 2004.

[8] A. Cooper, R. Reimann, D. Cronin and C. Noessel, About Face: The Essentials of Interaction Design. John Wiley and Sons, 2014.

[9] P. G. Armour, The Five Orders of Ignorance. Communications of the ACM, 43(10): 17-20, 2000.

[10] S. A. Fricker, C. Thümmler and A. Gavras, Requirements Engineering for Digital Health. Springer International Publishing, 2015.