

Integrating Software and Usability Engineering through Jointly-constructed, Event-based Stories

John Teofil Paul Nosek

Temple University
Rm. 316 Wachman Hall
Philadelphia, PA 19122
215-204-7232

nosek@temple.edu

ABSTRACT

This position paper proposes that event-based stories appear to have the potential to provide a simple, but powerful technique for users and developers to communicate emotional and informational needs, redesign processes, and structure the user interface design within the agile development paradigm. Informal evaluation of the use of event-based stories in several development projects suggest that event-based stories could be useful in integrating software and usability engineering. Controlled experiments, in addition to more formal case analyses are the next steps.

Categories and Subject Descriptors

D.2.2 Design Tools and Techniques H.5.2 User Interfaces

General Terms

Design, Human Factors

Keywords

Events, stories, scenarios, usability engineering, software engineering

1. POSITION PAPER

Software engineering has focused on functionality, i.e., the system must do “x” [2]. However, much of software development focuses on usability issues [7]. Usability Engineering focuses more on how easy the developed system is to learn and use, but these divisions are artificial. The earlier user feedback begins and the more it can be maintained throughout the development process, the better [3]. Nosek & Ahrens [4], Nosek & Schwartz [5], Nosek & Roth [6] explored techniques that can be used by users and developers over the translation process from problem statement to developed system. Such techniques must be powerful enough for users to express needs that can be ultimately translated into code by developers. Through experience, end-users have not found most technically-oriented techniques, such as data flow

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

I-USED'08, September 24, 2008, Pisa, Italy

diagrams, easy to learn and use. Juristo et al recommend applying elicitation patterns to garner usability requirements “after a preliminary version of the software requirements has been created [1].” This position paper explores the use of jointly-constructed, event-based stories as a powerful, flexible communications technique among end users and developers. Stories can be employed from the earliest stages of development and in concert with applying other techniques, such as, elicitation patterns. For example, agile-based development recognizes the real-world demands that development work be divided in time-segmented portions of completed deliverables, which includes, code, testing, and interface design. Rosson and Carroll use scenarios and a process of refinement of the scenarios from problem to activity design to information design to interaction design [7]. However, they lack sufficient granularity to easily identify the problems and track the refinement of the scenarios from problem description through interaction design. This is because a single scenario as employed by Rosson and Carroll can include multiple events and activities and incorporate wordier, less directly relevant task descriptions that may make for a more interesting story but adds complexity and reduces clarity. Events help to organize the problem space for software development and the construction of stories based on these events may help to integrate software and usability engineering. Developing stories for single events provides finer granularity and makes it easier to track refinements of the scenarios.

Information-based techniques by their nature filter out any emotional aspects discovered in the initial information gathering process. Through the initial story of the problem statement, users may be able to place themselves within the story and judge whether the developer understands both emotional and informational aspects. For example in the sample story below, the user can observe that the developer has incorporated the emotion of worry in the problem scenario and the reduction of this worry in the activity design. Emotions add strength or importance to a situation. Specifically recognizing emotions within stories validates the user’s contribution and may make the user more confident that the developer accurately understands the situation. Users that can read have the necessary capabilities to modify, and therefore, should be able to co-construct the stories without additional training in any particular technically-oriented technique.

In the next phase, developers can incorporate process redesign in refining the problem statement to incorporate the new activities with the proposed system. This process can be refined through information and interaction design. Figure 1 shows how Event-based stories can proceed in tandem with technically-oriented

techniques, which focus on coding and testing. Figure 2 shows how the solution space can be subdivided by Event-based stories and Technically-oriented techniques. An example is given for how stories may be refined around a single event.

Event 1: Prof. Bob London missed the flight after a conference and so had to cancel the class next day. (Instructor alters the class schedule)

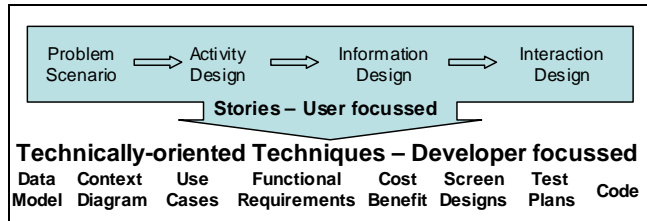


Figure 1: Event-based Stories in Software and Usability Eng.

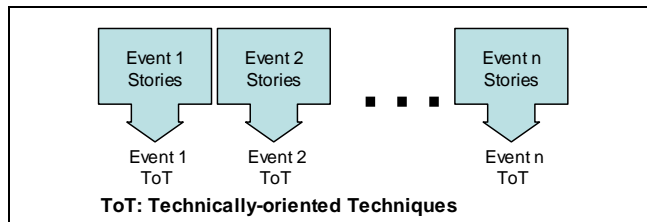


Figure 2: Division of Solution Space by Events

Problem Scenario for Event 1: Prof. Bob London was in Houston, TX on Tuesday for a conference and missed the last flight. He realized that he would miss the class next morning and was worried that students would show up to class confused and angry at him. However, he had neither the list of students for the class nor their contact information with him. So he sent an email to Ms. Tika Farrell, the dept.'s secretary, asking her to let the students know that the class for Wednesday morning is cancelled. Ms. Tika was annoyed with having to do one more thing and didn't have the contact information for the students. So she replied saying that she will post a note in the class room. Prof. B L searched through his emails and found a student's email. He made an educated guess that the student was in the Wednesday morning class and emailed him saying that the Wednesday class is cancelled and that he let other students know. It was already late in Philadelphia and Prof. B L was not sure if the student read his email that night.

(Process Redesign) Activity Design Scenario from Problem Scenario for Event 1: (same as above ...) He realized that he would miss the class next morning. B L was not worried and didn't have to bother the secretary. He connected his laptop to the Internet and logged into the Online Instructional Support System and sent out an announcement to the students to the effect that the class is cancelled. Wednesday morning, Prof. B L checked the system and found that 12 out of 15 students read the announcement. He easily sent a reminder to the 3 students who didn't read the announcement.

Information Design Scenario from Activity Design Scenario for Event 1: (same as above ...) Prof. B L connected his laptop to the Internet and logged into the Online Instructional Support

System. He noticed that there were a couple of alerts, which he decided to ignore for the time being. Prof. B L had predefined groups of students in various courses. These were in his personal address book. He started to compose a new 'announcement'. A window similar to composing an email got displayed. Prof. B L typed in the announcement and sent it to the predefined group of students in his Wednesday class. Wednesday morning, Prof. B L checked the system and found that 12 out of 15 students read the announcement.

Interaction Design Scenario from Activity Design Scenario for

Event 1: (same as above - interaction refinements are underlined) Prof. B L connected his laptop to the Internet and logged into the Online Instructional Support System. He noticed that there were a couple of alerts, which he decided to ignore for the time being. So, he clicked on the "remind me later" button. The main menu showed up. Prof. B L selected "messaging" option. He selects the "new announcement" item. A window similar to composing an email shows up. In the compose window, he selected the "To" field; right clicked and selected "predefined groups". The predefined groups in his personal address book showed up. He selected the group that corresponded to the students in his Wednesday class. The "To" field got populated with the group information. He typed in the announcement information in the "message" field and pressed the "send" button to send the announcement.

2. SUMMARY

This position paper proposes that event-based stories appear to have the potential to provide a simple, but powerful technique for users and developers to communicate emotional and informational needs, redesign processes, and structure the user interface design within the agile development paradigm. Informal evaluation of the use of event-based stories in several development projects suggest that event-based stories could be useful in integrating software and usability engineering. Controlled experiments, in addition to more formal case analyses are the next steps.

3. ACKNOWLEDGMENTS

George Mathew developed the initial story example. I thank the anonymous reviewers for their thoughtful and constructive comments.

4. REFERENCES

[1] Juristo, N., Moreno, A. M., Sanchez-Segura, M. (2007), "Guidelines for Eliciting Usability Functionalities", IEEE Transactions on Software Engineering, Vol. 33, No. 11, November 2007, pp. 744-758.

[2] Larman, C. (2004) Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design, Prentice Hall, New York.

[3] Nosek, J.T., Sherr, D. (1984) "Getting the Requirements Right vs. Getting the System Working - Evolutionary Development," in Bemelmans, T.M.A. (ed.) Beyond Productivity: Information Systems Development for Organizational Effectiveness, North Holland, New York.

[4] Nosek, J.T., Ahrens, J. (1986), "An Experiment to Test User Validation of Requirements: Data Flow Diagrams v.

Task-oriented Menus" (with J. Ahrens), International Journal of Man-Machine Studies, Vol. 25, No 6, December, 675-684.

[5] Nosek, J.T., Schwartz, R. (1988), "User Validation of Information System Requirements: Some Empirical Results", IEEE Transactions on Software Engineering, Vol. 14, No. 9, September, 1372-1375.

[6] Nosek, J.T., Roth, I. (1990), "A Comparison of Formal Knowledge Representation Schemes as Communication

Tools: Predicate Logic vs. Semantic Network", International Journal of Man-Machine Studies, Vol. 33, 227-239.

[7] Rosson, M.B. & Carroll, J.M.(2002) Usability Engineering: Scenario-based Development of Human Computer Interaction, Morgan Kaufmann Publishers Inc., San Francisco, CA.