

# Integrating GUI Mockups and User Stories

Olga Liskin and Tobias Baum  
Leibniz University Hannover, Germany  
{olga.liskin, tobias.baum}@inf.uni-hannover.de

## 1 Context and Problem

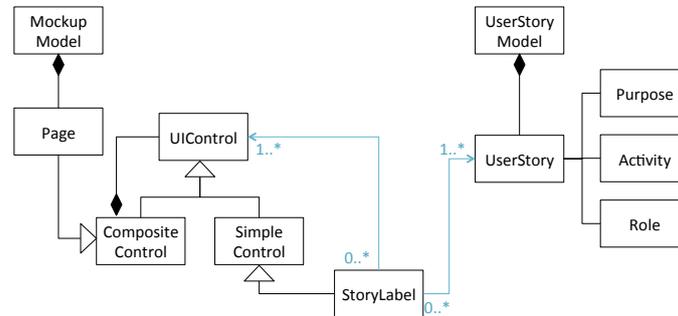
An important goal in requirements engineering is to enable good requirements communication among all project participants. Especially in projects with a variety of stakeholders, it is important to repeatedly collect and discuss their different views and requirements throughout the whole project. However, most of the stakeholders have other main tasks and only little time to spend on requirements analysis and documentation. Therefore, it is important to enable quick and efficient ways to gain viable requirements artifacts. Especially in iterative or agile approaches, stakeholders are in demand to actively contribute to requirements analysis and documentation. Two frequent types of artifacts used in such situations are user stories and GUI mockups.

GUI mockups have proven to be particularly supportive for discussions about requirements details. Stakeholders with little experience in requirements engineering find GUI mockups tangible and with their help can express their requirements more easily [1]. In order to eventually convert the acquired information into story cards, additional effort and a switch of context are necessary. Often, one needs to switch back and forth between different views, which slows down the whole activity. Alternatively, verbal comments are written down as notes, which puts them at the risk of being implemented directly and never documented in a requirements artifact [1].

## 2 Principal Ideas

We present a concept that integrates user stories into GUI mockups. Special elements – which we call *Story Labels* – are inserted directly into a mockup view to complement its elements with information about user requirements. From these labels, user stories are generated and automatically inserted into the project backlog. Likewise, information from the backlog can be visualized directly inside GUI mockups. In this way, the constant switch of context can be avoided and allows the project participants to discuss and plan directly with user-understandable mockups.

The concept is realized by merging the models of mockups and user stories, and integrating the respective views. Figure 1 depicts the integrated meta-model, based on a GUI mockup meta-model by Rivero et al. [2] and a user story meta-model which can, for example, be extracted from [3]. An extra element *StoryLabel* links both elements. It is associated to one or more *UIControls* and then described in more details by one or more *UserStories*. The *StoryLabel* itself is a *SimpleControl* containing a description, which – like other GUI elements – has a position and is visible in the mockup.



**Figure 1.** Metamodel for integration of GUI mockups and user stories

For example, when discussing an online bookstore system, a stakeholder can create a *StoryLabel* for a GUI element with the author names, saying “*make the author name clickable and then display all books by that author*”. The stakeholder can then enhance the label. For the activity part, she leaves the default value, which is the label’s description. She adds “*customer*” as a role and “*so that I can find similar books by that author more quickly*” as a purpose and hereby has created a user story.

**Benefits:** An integrated environment for engineering GUI mockups and user stories could facilitate the following benefits:

*The user can enter his or her requirements with less effort.* The user does not need to switch between GUI mockups and a backlog to enter requirements. The system can suggest suitable existing user stories as an alternative to creating a new story.

*The user can create requirements from remarks emerging in the discussion of mockups.* A story’s activity element is created from the label description. It can be improved later. Labels and stories are automatically transferred into the backlog environment. The GUI mockup and label are automatically attached to the linked story.

*The developers can check completeness of the requirements.* The system displays which elements are not associated with a story or do not have a defined purpose yet.

*Developers and users can use the supplementary information for planning and prioritization based on GUI mockups.* For the GUI elements, the system can display and aggregate estimated effort and assigned iterations.

Currently, we are developing a prototype that implements the concept. We plan to use it to evaluate the concept’s feasibility and the relevance of the benefits.

## References

1. Liskin, O.: How artifacts support and impede requirements communication. Accepted for publication in REFSQ 2015, LNCS. Springer International Publishing, 2015.
2. Rivero, J.M., Rossi, G., Grigera, J., Burella, J., Luna, E. R.: From mockups to user interface models: An extensible model driven approach. In Current Trends in Web Engineering. LNCS, vol. 6385, pp. 13–24, Springer Berlin Heidelberg, 2010.
3. Cohn, M.: User Stories Applied. Addison Wesley, 2004.