Universitat de Lleida

Requirements Engineering

User Stories

Especialització en Enginyeria de Programari
Grau en Enginyeria Informàtica
Escola Politècnica Superior

Marta Oliva

Universitat de Lleida

User Stories

"… the design of the solution has to be communicated. Most iterative processes use User Stories for this purpose. A collection of user stories represents the functionality needed for the next release."

[Robertson & Robertson 2012]

Outline

- What is a User Story?
- Write Good User Stories
- User Role Modeling
- Gathering Stories
- Working with User Proxies
- Acceptance Testing User Stories
- Guidelines for Good Stories
- What Stories Are Not

Outline

- What is a User Story?
- Write Good User Stories
- User Role Modeling
- Gathering Stories
- Working with User Proxies
- Acceptance Testing User Stories
- Guidelines for Good Stories
- What Stories Are Not

What is a User Story?

- User Stories are a way of representing requirements.
- The usual form – at least the suggested – of a user story is this:

As a [role], I want [feature] so that [reason]

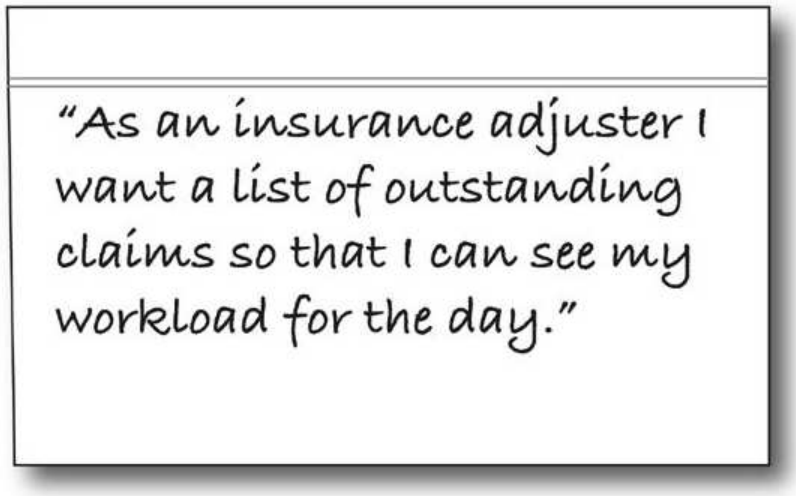


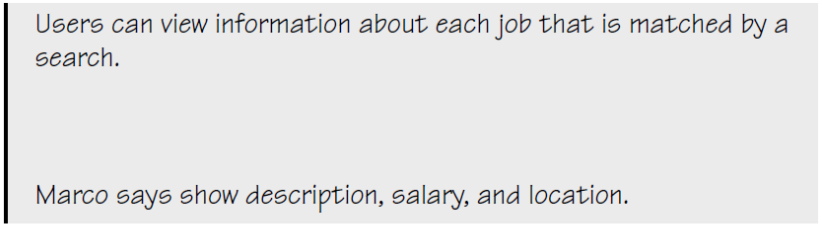
Figure 14.2. A user story written on an index card.

- ✓ The user story is initially handwritten on an index card. It is brief, and at this stage of its evolution, is really a placeholder signaling that there is some functionality whose details have yet to be worked out.

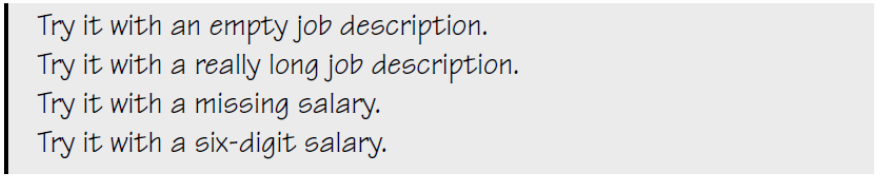
[Robertson & Robertson 2012]

What is a User Story?

- According Cohn, a user story describes functionality that will be valuable to either a user or purchaser of a system or software. User stories are composed of three aspects:
  - a written description of the story used for planning and as a reminder
  - conversations about the story that serve to flesh out the details of the story
  - tests that convey and document details and that can be used to determine when a story is complete



Story Card 1.2 A story card with a note.



Story Card 1.3 The back of a story card holds reminders about how to test the story.

[Cohn 2004]

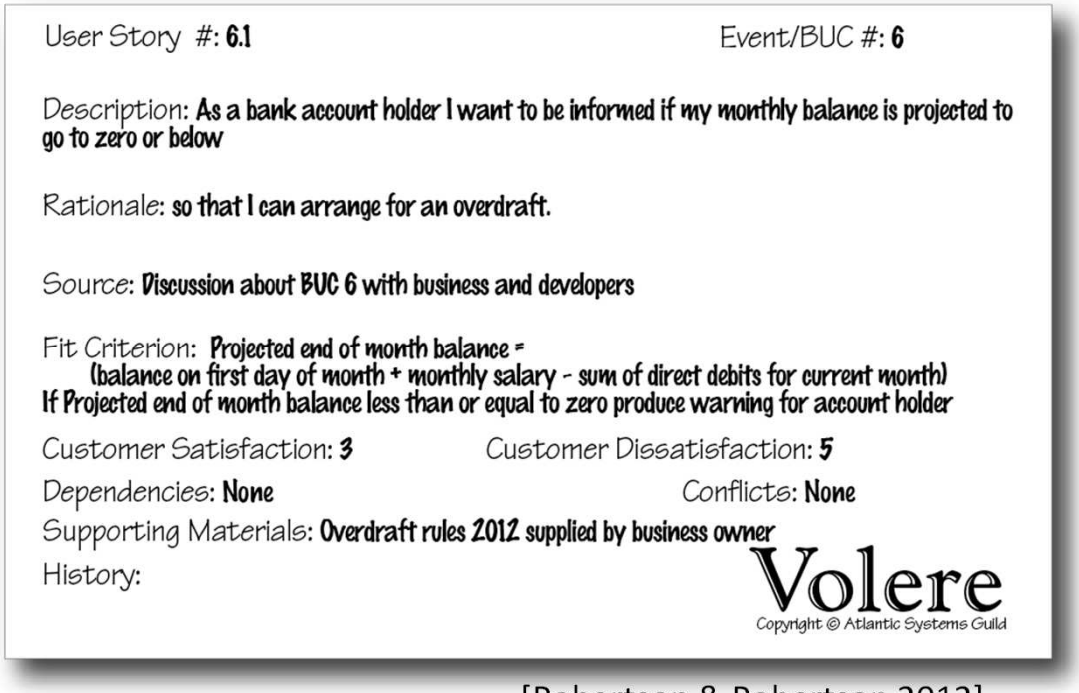
Outline

- What is a User Story?
- Write Good User Stories
- User Role Modeling
- Gathering Stories
- Working with User Proxies
- Acceptance Testing User Stories
- Guidelines for Good Stories
- What Stories Are Not

Write Good User Stories

- The level of granularity of user stories varies; but it is normally somewhere between, but might be either, a product use case (PUC) and an atomic requirement.

✓ Most iterative development teams write their stories on blank cards, but you can also use the *Volere snow card* for this purpose.


User Story #: 6.1 Event/BUC #: 6

Description: As a bank account holder I want to be informed if my monthly balance is projected to go to zero or below

Rationale: so that I can arrange for an overdraft.

Source: Discussion about BUC 6 with business and developers

Fit Criterion: Projected end of month balance =
(balance on first day of month + monthly salary - sum of direct debits for current month)
If Projected end of month balance less than or equal to zero produce warning for account holder

Customer Satisfaction: 3 Customer Dissatisfaction: 5

Dependencies: None Conflicts: None

Supporting Materials: Overdraft rules 2012 supplied by business owner

History:

Volere

Copyright © Atlantic Systems Guild

[Robertson & Robertson 2012]

Write Good User Stories

- A story card contains a short description of user- or customer-valued functionality.
- A story card is the visible part of a story, but the important parts are the conversations between the customer and developers about the story.
- The customer team includes those who ensure that the software will meet the needs of its intended users. This may include testers, a product manager, real users, and interaction designers.
- The customer team writes the story cards because they are in the best position to express the desired features and because they must later be able to work out story details with the developers and to prioritize the stories.
- Stories are prioritized based on their value to the organization.
- Releases and iterations are planned by placing stories into iterations.

[Cohn 2004]

Write Good User Stories

- Velocity is the amount of work the developers can complete in an iteration.
- The sum of the estimates of the stories placed in an iteration cannot exceed the velocity the developers forecast for that iteration.
- If a story won't fit in an iteration, you can split the story into two or more smaller stories.
- Acceptance tests validate that a story has been developed with the functionality the customer team had in mind when they wrote the story.
- User stories are worth using because they emphasize verbal communication, can be understood equally by you and the developers, can be used for planning iterations, work well within an iterative development process, and because they encourage the deferring of detail.

[Cohn 2004]

Write Good User Stories

- Ideally, stories are independent from one another. This isn't always possible but to the extent it is, stories should be written so that they can be developed in any order.
- The details of a story are negotiated between the user and the developers.
- Stories should be written so that their value to users or the customer is clear. The best way to achieve this is to have the customer write the stories.
- Stories may be annotated with details, but too much detail obscures the meaning of the story and can give the impression that no conversation is necessary between the developers and the customer.
- One of the best ways to annotate a story is to write test cases for the story.
- It is important for developers to be able to estimate the size of a story or the amount of time it will take to turn a story into working code. If they are too big, compound and complex stories may be split into multiple smaller stories. If they are too small, multiple tiny stories may be combined into one bigger story.

[Cohn 2004]

- Stories need to be testable.

Outline

- What is a User Story?
- Write Good User Stories
- User Role Modeling
- Gathering Stories
- Working with User Proxies
- Acceptance Testing User Stories
- Guidelines for Good Stories
- What Stories Are Not

User Role Modeling

- Most project teams consider only a single type of user. This leads to software that ignores the needs of at least some user types.
- To avoid writing all stories from the perspective of a single user, identify the different user roles who will interact with the software.
- By defining relevant attributes for each user role, you can better see the differences between roles.
- Some user roles benefit from being described by *personas*. Remember that a *persona* is an imaginary representation of a user role. The persona is given a name, a face, and enough relevant details to make them seem real to the project members.
- For some applications, extreme characters may be helpful in looking for stories that would otherwise be missed. So, system designers should consider users with exaggerated personalities.

[Cohn 2004]

User Role Modeling

Role Modeling Steps

We will use the following steps to identify and select a useful set of user roles:

1. brainstorm an initial set of user roles
2. organize the initial set
3. consolidate roles
4. refine the roles

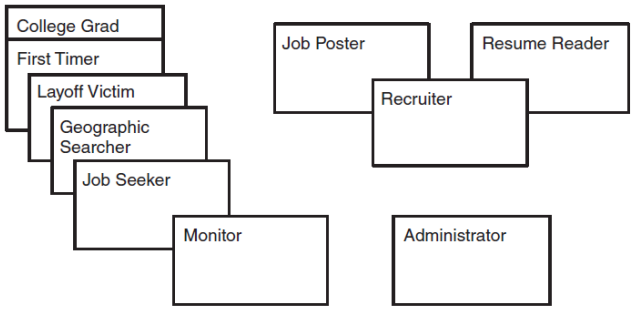


Figure 3.1 *Organizing the user role cards on a table.*

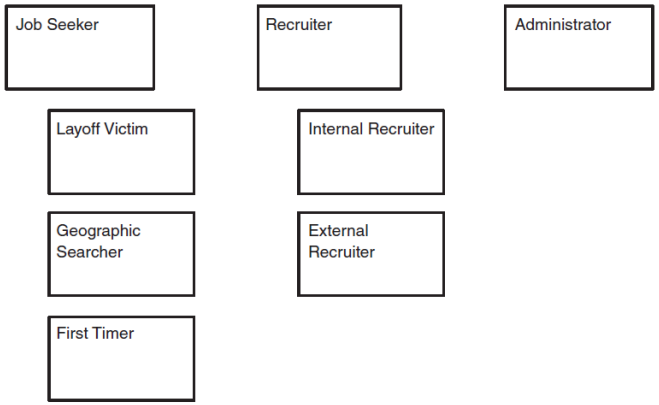


Figure 3.2 *The consolidated role cards.*

User Role: Internal Recruiter

Not particularly computer-savvy but quite adept at using the Web. Will use the software infrequently but intensely. Will read ads from other companies to figure out how to best word her ads. Ease of use is important, but more importantly what she learns must be easily recalled months later.

Figure 3.3 *A sample user role card.*

[Cohn 2004]

Outline

- What is a User Story?
- Write Good User Stories
- User Role Modeling
- Gathering Stories
- Working with User Proxies
- Acceptance Testing User Stories
- Guidelines for Good Stories
- What Stories Are Not

Gathering Stories

- The idea of eliciting and capturing requirements is wrong. It leads to the twin fallacies that users already know all the requirements and that requirements can be captured and locked in a cage where they will remain unchanged.
- The metaphor of trawling for requirements is far more useful: it captures the ideas that there are different sizes of requirements, that requirements may change over time, and that it takes skill to find the requirements.
- While agile processes are supportive of requirements that emerge late in the process, you should still start by looking forward to approximately the end of the intended release and write the user stories you can easily see.
- User stories can be found by interviewing users, observing users, questionnaires, and holding story-writing workshops.
- The best results are achieved by using a combination of methods rather than overreliance on any one method.
- The most useful answers are given in response to open-ended, context-free questions, such as “Tell me about how you’d like to search for a job” rather than “Will you search for a job by title?” [Cohn 2004]

Outline

- What is a User Story?
- Write Good User Stories
- User Role Modeling
- Gathering Stories
- Working with User Proxies
- Acceptance Testing User Stories
- Guidelines for Good Stories
- What Stories Are Not

Working with User Proxies

- It is vital that a project include one or more real users on the customer team. Unfortunately, it is often difficult to get the users we need. When we cannot get as many users as we want to represent different perspectives of the product, we need to resort to *user proxies*.
- Although not ideal, it is still possible to write great software with a user proxy rather than a real user. There are different types of user proxies.
- The users' manager may not be an appropriate user proxy unless she/he is also a user.
- Development managers make tempting user proxies because they are already involved in the day-to-day detail of the project. However, the development manager is rarely an intended user of the software being built and is therefore a poor choice as a user proxy.
- In product companies, the customer frequently comes from the marketing group. Someone from the marketing group is often a good choice as a user proxy but must overcome the temptation to focus on the quantity rather than quality of features.

[Cohn 2004]

Working with User Proxies

- Salespeople can make good customers when they have contact with a broad variety of customers who are also users. Salespeople must avoid the temptation to focus on whatever story could have won the last lost sale. In all cases, salespeople make excellent conduits to users.
- Domain experts can make excellent user proxies but must avoid the temptation to write stories for a product that only someone with their expertise can use.
- Customers, those who make the purchasing decision, can make great user proxies if in close communication with the users for whom they are purchasing the software. Obviously, a customer who is also a user is a fantastic combination.

[Cohn 2004]

Working with User Proxies

- In order to be good user proxies, trainers and technical support personnel must avoid the temptation to focus too narrowly on the aspects of the product they see every day.
- There are some techniques for working with user proxies, including:
  - the use of user task forces, when user exists but access is limited,
  - using multiple user proxies, when there really is no user available,
  - competitive analysis, when you are developing software that will compete with other commercial products, and
  - releasing early to get user feedback, even if the release is called preliminary or an early beta, getting it into the hands of users early will help identify inconsistencies between the thinking of your user proxy and your real users.

[Cohn 2004]

Outline

- What is a User Story?
- Write Good User Stories
- User Role Modeling
- Gathering Stories
- Working with User Proxies
- Acceptance Testing User Stories
- Guidelines for Good Stories
- What Stories Are Not

Acceptance Testing User Stories

- Acceptance tests are used to express details that result from conversations between a customer and a developer.
- Acceptance tests document assumptions about the story a customer has that may not have been discussed with a developer.
- Acceptance tests provide basic criteria that can be used to determine if a story is fully implemented.
- Acceptance tests should be written by the customer rather than by a developer.
- Acceptance tests are written before the programmer begins coding.
- Stop writing tests when additional tests will not help clarify the details or intent of the story.

[Cohn 2004]

Acceptance Testing User Stories

- One excellent tool for automating acceptance tests is Ward Cunningham's Framework for Integrated Test (available at fit.c2.com), or FIT for short. Using FIT, tests are written in a familiar spreadsheet or tabular format.
- FitNesse (which uses FIT) (available from fitnesse.org) is rapidly becoming a very popular approach for writing acceptance tests on agile projects. Because tests are expressed in spreadsheet-like tables within web pages, the effort for customers to identify and write tests is greatly reduced.

Table 6.1 *Testing for valid credit cards with a table that can be used by FIT and FitNesse.*

CardType	Expiration	Number	valid()
Visa	05/05	4123456789011	true
Visa	05/23	4123456789012349	false
MasterCard	12/04	5123456789012343	true
MasterCard	12/98	5123456789012345	false
MasterCard	12/05	42	false
American Express	4/05	341234567890127	true

[Cohn 2004]

Acceptance Testing User Stories

Types of Testing

There are many types of testing, and the customer and development team should work together to ensure that the appropriate types of testing are occurring. For most systems, story testing is largely functional testing, which ensures that the application functions as expected. There are, however, other types of testing to be considered. For example, you may want to consider any or all of the following:

- User interface testing, which ensures that all of the components of the user interface behave as expected
- Usability testing, which is done to ensure an application that can be easily used
- Performance testing, which is done to gauge how well the application will perform under various workloads
- Stress testing, in which the application is subjected to extreme values of users, transactions, or anything else that may put the application under stress.

[Cohn 2004]

Outline

- What is a User Story?
- Write Good User Stories
- User Role Modeling
- Gathering Stories
- Working with User Proxies
- Acceptance Testing User Stories
- Guidelines for Good Stories
- What Stories Are Not

Guidelines for Good Stories

- To identify stories, start by considering the goals of each user role in using the system.
- When splitting a story, try to come up with stories that cut through all layers of the application.
- Try to write stories that are of a size where the user feels justified in taking a coffee break after completing the story.
- Augment stories with other requirements gathering or documenting techniques as necessary for the project's domain and environment.
- Create constraint cards and either tape them to a shared wall or write tests to ensure the constraints are not violated.

> The system must support peak usage of up to 50 concurrent users.
>
> Constraint

Story Card 7.1 An example of a constraint story card.

[Cohn 2004]

Guidelines for Good Stories

- Write smaller stories for functionality the team will implement soon, and write broad, high-level stories for functionality further into the future.
- Keep the user interface out of the stories for as long as possible.

> Print dialog allows the user to edit the printer list. The user can add or remove printers from the printer list. The user can add printers either by auto-search or manually specifying the printer DNS name or IP address. An advanced search option also allows the user to restrict his search within specified IP addresses and subnet range.

Story Card 7.2 A card with too much user interface detail.

- When practical, include the user role when writing the story.

[Cohn 2004]

Guidelines for Good Stories

- Write stories in active voice. For example, say “A Job Seeker can post a resume” rather than “A resume can be posted by a Job Seeker.”
- Write stories for a single user. Instead of “Job Seekers can remove resumes” write “A Job Seeker can remove her/his own resumes.”
- Have the customer, rather than a developer, write the stories.
- Keep user stories short, and don’t forget their purpose as reminders to hold conversations.
- Don’t number story cards.

[Cohn 2004]

Outline

- The Need for Iterative Development
- An Iterative Requirements Process
- Business Value Analysis and Prioritization
- Iterative Requirements Roles
- What is a User Story?
- Write Good User Stories
- User Role Modeling
- Gathering Stories
- Working with User Proxies
- Acceptance Testing User Stories
- Guidelines for Good Stories
- What Stories Are Not

What Stories Are Not

- User stories are different from IEEE 830 software requirements specifications, use cases and interaction design scenarios.
- No matter how much thinking, thinking and thinking we do, we cannot fully and perfectly specify a non-trivial system upfront.
- There is a valuable feedback loop that occurs between defining requirements and users having early and frequent access to the software.
- It is more important to think about users' goals than to list the attributes of a solution.
- User stories are similar to a use case scenario. But use cases still tend to be larger than a single story and can be more prone to containing embedded assumptions about the user interface.

[Cohn 2004]

What Stories Are Not

- Additionally, user stories differ from use cases in their completeness and longevity. Use cases are much more complete than are user stories. Use cases are designed to be permanent artifacts of the development process; user stories are more transient and not intended to outlive the iteration in which they are developed.
- User stories and use cases are written for different purposes. Use cases are written so that developers and customers can discuss them and agree to them. User stories are written to facilitate release planning and to serve as reminders to fill in requirements details with conversations.
- Unlike IEEE 830 specifications and use cases, user stories are not artifacts of analysis activities. Rather, user stories are a support tool for analysis.
- Interaction design scenarios are much more detailed than user stories and differ in the kind of detail provided.
- A typical interaction design scenario is much larger than a user story. One scenario may comprise multiple use cases, which, in turn, may comprise many user stories.

[Cohn 2004]

Universitat de Lleida

Bibliografia

- **S. Robertson & J. Robertson**. *Mastering the Requirements Process: Getting Requirements Right* (3rd edition), Addison-Wesley, 2012.
- **M. Cohn**. *User Stories Applied: For Agile Software Development*, Addison Wesley, 2004.