

Dad, tell me a user story

How to write user stories and be happy

Federico Gobbo Andrea Gervasini

federico.gobbo@uninsubria.it; info@agilmente.eu

Università dell'Insubria, Varese; Agilmente snc, Varese

(cc) Some rights reserved.

ESSAP2008 Varese, 2008 July the 2nd



What is a User Story?

User Story was coined by Kent Beck in *eXtreme Programming Explained* to refer to the small bits of work used by the agile practitioners to plan development.



What is a User Story?

User Story was coined by Kent Beck in *eXtreme Programming Explained* to refer to the small bits of work used by the agile practitioners to plan development.

User stories are written from the users perspective, with the domain language of the customer.



What is a User Story?

User Story was coined by Kent Beck in *eXtreme Programming Explained* to refer to the small bits of work used by the agile practitioners to plan development.

User stories are written from the users perspective, with the domain language of the customer.

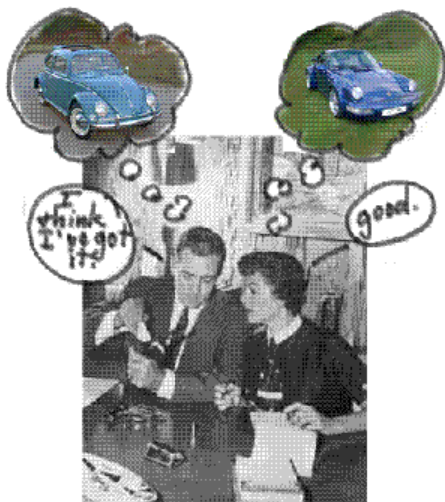
This workshop is a derivative work of Agile Requirements Exploration presented in XP Days BE 2007 by Dave Nicolette and Brett Schuchert and User Story Mapping by Jeff Patton presented in XP 2008.



The common goal is to build a shared understanding...



...but sometimes misunderstandings do happen



User stories are concise and lightly structured

User Stories are written on index cards, so to maintain them concise, and to reduce the impact of mistakes.



User stories are concise and lightly structured

User Stories are written on index cards, so to maintain them concise, and to reduce the impact of mistakes.

A user story usually has:

- a title
- a concise narrative (recto)
- the acceptance tests of the narrative (verso)



The typical pattern of narratives (recto)

- As a (*type of user*)
- I want to (*perform some task*)
- so that I can (*reach some goal*)



The typical pattern of narratives (recto)

- As a (*type of user*)
- I want to (*perform some task*)
- so that I can (*reach some goal*)

There are other patterns in use – e.g. using personas – but we won't use them here. Let us be familiar with the most used, at first.



What's acceptance tests? (verso)

Acceptance tests are written by the customers and the developers so all people involved know exactly when the story is complete.

Acceptance tests are black box system tests.



What's acceptance tests? (verso)

Acceptance tests are written by the customers and the developers so all people involved know exactly when the story is complete.

Acceptance tests are black box system tests.

Acceptance tests are also regression tests prior to a production release.



What's acceptance tests? (verso)

Acceptance tests are written by the customers and the developers so all people involved know exactly when the story is complete.

Acceptance tests are black box system tests.

Acceptance tests are also regression tests prior to a production release.

Typically, they describe the possible endings (outputs) of the narrative.



An example: view a product's location (recto)

- *As a harried shopper*
- *I want to view a product's location*
- *so that I can find it and buy it quickly*

example by Jeff Patton, XP2008



Possible acceptance tests (verso)

Success: harried shopper finds the product's location

- 'remember me' ticked – store cookie/automatic login next time.
- 'remember me' not ticked - force login next time.



Possible acceptance tests (verso)

Success: harried shopper finds the product's location

- 'remember me' ticked – store cookie/automatic login next time.
- 'remember me' not ticked - force login next time.

Failure: harried shopper don't find the product's location. Display message:

- "Sorry, your product isn't in stock now, please try again later"
- "Perhaps did you mean (*different spelling*)?"
- "Ooops! Service unavailable, please try again later"
- "Your account has expired. Please register again"



Caveat: User Stories \neq Use Cases

Use Cases are used in the OO-software development approach and depicts the interaction between agents. In use cases the narratives describe what agents (humans and machine alike) do step by step.

Use cases imply design decisions, in particular the use of a specific software tool.



Caveat: User Stories \neq Use Cases

Use Cases are used in the OO-software development approach and depicts the interaction between agents. In use cases the narratives describe what agents (humans and machine alike) do step by step.

Use cases imply design decisions, in particular the use of a specific software tool.

On the contrary, *user stories focus on tasks, not tools.*



The lifecycle of User Stories

- **writing** Customers and developers write the initial narrative
- **conversation** Re-writing, splitting, refining the narrative
- **confirmation** Acceptance tests writing so that the story can be signed when completed



A set of User Stories is a Product Backlog

Typically a certain number of user stories are written in a planning game session. They are prioritized according to:



A set of User Stories is a Product Backlog

Typically a certain number of user stories are written in a planning game session. They are prioritized according to:

- the **business value**, given by the customer(s)
- the **effort score**, given by the developer team



Ideal features of User Stories

- **Indipendent** from the structure/content/layout division
- **Negotiable** the scope is not fixed: discuss details!
- **Valuable** from the customers' perspective
- **Estimable** easy to figure our how much work¹
- **Small** from the developer team perspective
- **Testable** i.e. verifiable by both customers and developers



¹otherwise, do a spike

Ideal features of User Stories

- **Indipendent** from the structure/content/layout division
- **Negotiable** the scope is not fixed: discuss details!
- **Valuable** from the customers' perspective
- **Estimable** easy to figure our how much work¹
- **Small** from the developer team perspective
- **Testable** i.e. verifiable by both customers and developers

INVEST in your User Stories!



¹otherwise, do a spike

Now, the rules of the game



from the Italian Agile Day 2007 workshop
14/22



The actors

- You are divided into 3 teams in competition
- There is a Big Rich Customer (without cap)
- There is a User Story Teller (with the cap)
- The Big Rich Customer is not agile
- He gives the team the Solution made by his 'software analysts'



Some useful questions

- Who are the people who will be using the system?
- Why would they use it?
- What kind of activities might they do to meet their goals?
- What happens if they don't meet their goals? Who loses?
- Are there users who monitor and protect the interests of other users?

source: Jeff Patton, XP2008



The First Pomodoro

- Read the Solution
- Write a single User Story narrative in approx. 10 min.
- Discuss with the User Story Teller
- Propose your narrative to the Big Rich Customer
- Write with the Big Rich Customer the acceptance tests



The Second Pomodoro

- Write as more narratives you can in approx 10 min.
- Estimate the effort
- (You can ask to the User Story Teller for help)
- Ask for the business value to the Big Rich Customer
- Form a product backlog (i.e. a prioritized list)
- Put your backlog in the stories plastic album



The Third Pomodoro: inside trading

- Switch plastic albums between teams!
- Produce more user stories
- Rewrite your own user stories by heart in approx. 10 min.
- Form a new product backlog



The Fourth Pomodoro: acceptance tests

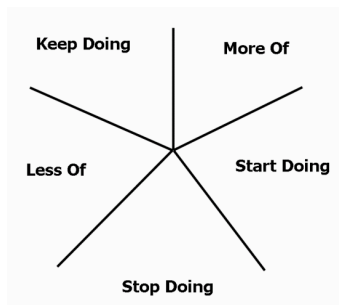
- The Big Rich Customer is no longer available
- We are the customer proxies (we were delegated to sign stories)
- Your velocity is 18 effort points
- Write acceptance tests



Who has won the game?



Thanks! How was your experience?!



Stick your comments on the starfish!



(cc) Federico Gobbo & Andrea Gervasini 2008. Published in Italy.

Attribution – Non commercial – Share alike 2.5

22/22

