

User Stories for Agile Requirements

Mike Cohn



Mike Cohn—background



- Programming for 20 years
- Author of
 - *User Stories Applied*
 - *Agile Estimating and Planning*
 - Java, C++, database programming books
- Founding member and director of the Agile Alliance and the Scrum Alliance
- Founder of Mountain Goat Software
 - Process and project management consulting and training

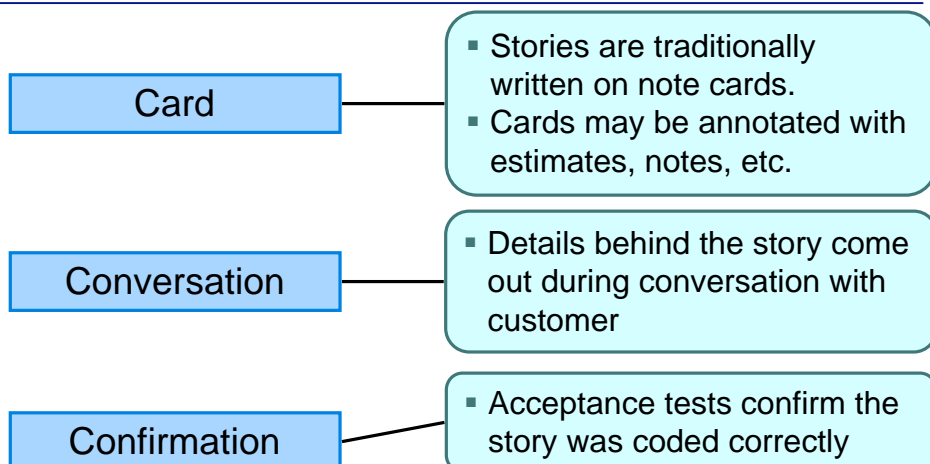
All slides copyright 2000-2005, Mountain Goat Software

Today's agenda

- What stories are
 - User role modeling
 - Story writing
 - INVEST in good stories
 - Why user stories?

All slides copyright 2000-2005, Mountain Goat Software

Ron Jeffries' Three Cs



All slides copyright 2000-2005, Mountain Goat Software

Samples – Travel reservation system

As a user, I can reserve a hotel room.

As a vacation planner, I can see photos of the hotels.

As a user, I can cancel a reservation.

As a user, I can restrict searches so I only see hotels with available rooms.

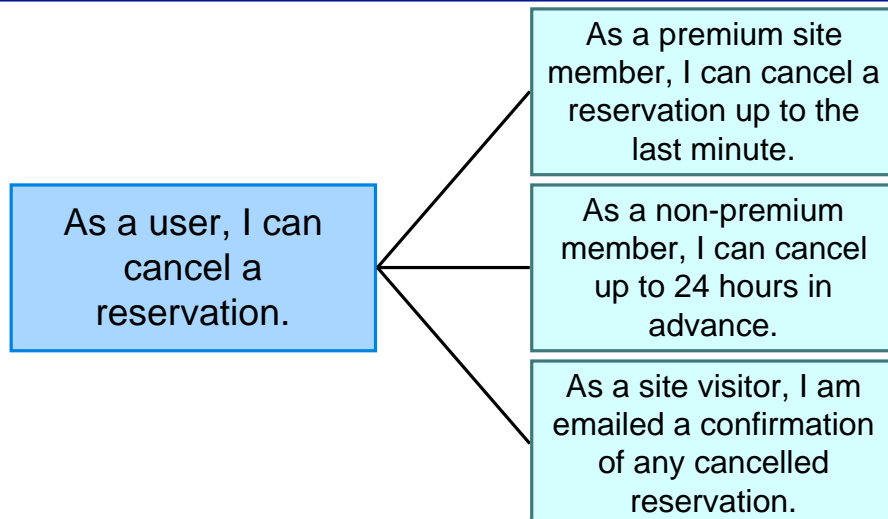
All slides copyright 2000-2005, Mountain Goat Software

Where are the details?

- As a user, I can cancel a reservation.
 - Does the user get a full or partial refund?
 - Is the refund to her credit card or is it site credit?
 - How far ahead must the reservation be cancelled?
 - Is that the same for all hotels?
 - For all site visitors? Can frequent travelers cancel later?
 - Is a confirmation provided to the user?
 - How?

All slides copyright 2000-2005, Mountain Goat Software

Details added in smaller “sub-stories”



All slides copyright 2000-2005, Mountain Goat Software

Details added as tests

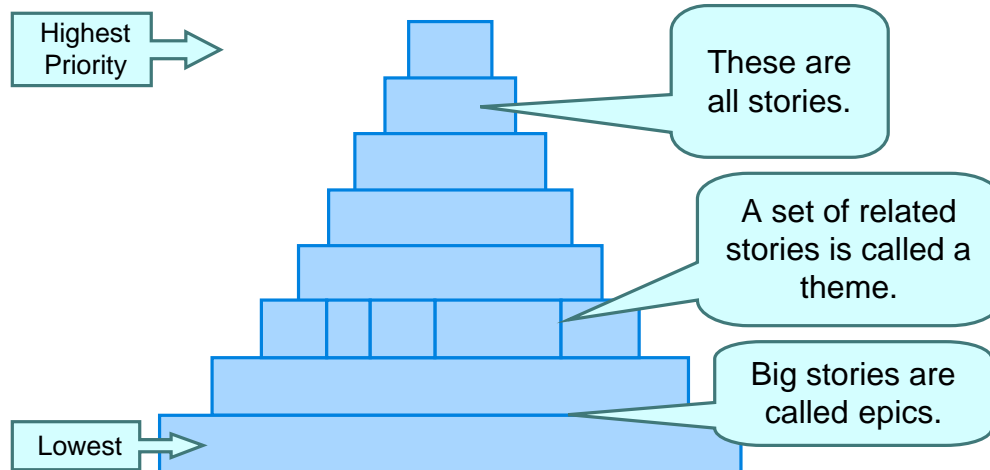
- High level tests are added to the story
 - Can be used to express additional details and expectations

As a user, I can cancel a reservation.

- Verify that a premium member can cancel the same day without a fee.
- Verify that a non-premium member is charged 10% for a same-day cancellation.
- Verify that an email confirmation is sent.
- Verify that the hotel is notified of any cancellation.

All slides copyright 2000-2005, Mountain Goat Software

Prioritized requirements list (PRL)



All slides copyright 2000-2005, Mountain Goat Software

Today's agenda

- What stories are
- User role modeling
- Story writing
- INVEST in good stories
- Why user stories?

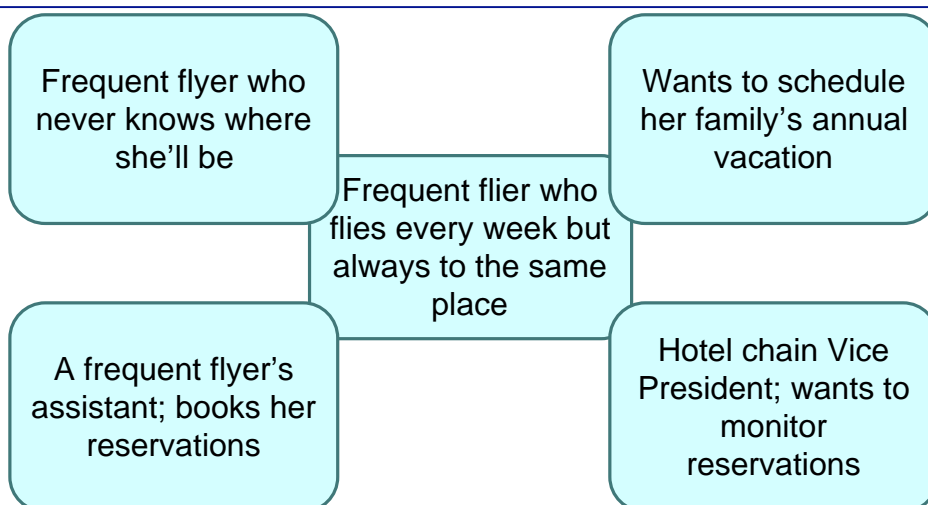
All slides copyright 2000-2005, Mountain Goat Software

“The User”

- Many projects mistakenly assume there’s only one user:
 - “The user”
- Write all stories from one user’s perspective
- Assume all users have the same goals
- Leads to missing stories

All slides copyright 2000-2005, Mountain Goat Software

Travel Site—Who’s the user?



All slides copyright 2000-2005, Mountain Goat Software

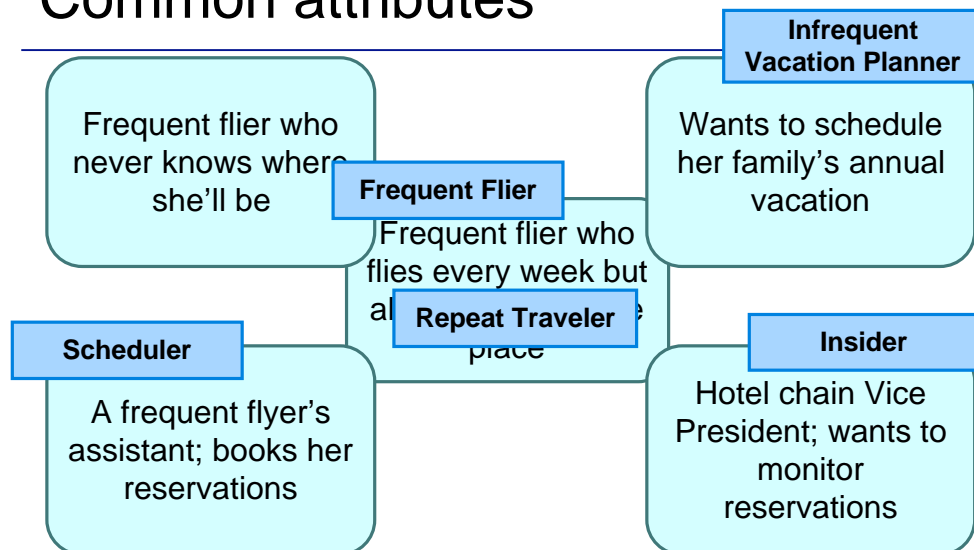
User roles

- Broaden the scope from looking at one user
- Allows users to vary by
 - What they use the software for
 - How they use the software
 - Background
 - Familiarity with the software / computers
- Used extensively in usage-centered design
- Definition
 - A user role is a collection of defining attributes that characterize a population of users and their intended interactions with the system.

Source: *Software for Use* by Constantine and Lockwood (1999).

All slides copyright 2000-2005, Mountain Goat Software

Common attributes



All slides copyright 2000-2005, Mountain Goat Software

Advantages of using roles

Users become tangible

Start thinking of software as solving needs of real people.

Avoid saying “the user”

Instead we talk about “a frequent flier” or “a repeat traveler”

Incorporate roles into stories

“As a <role>, I want <story> so that <benefit>.”

All slides copyright 2000-2005, Mountain Goat Software

Today's agenda

- What stories are
- User role modeling
- Story writing
- INVEST in good stories
- Why user stories?

All slides copyright 2000-2005, Mountain Goat Software

Questioning the users

“Would you like it in a browser?”

“Of course, now that you mention it!”

- A problem
 - The question is closed
 - {Yes | No}

All slides copyright 2000-2005, Mountain Goat Software

We can do better

“What would you think of having this app in a browser rather than as a native Windows application, even if it means reduced performance, a poorer overall user experience, and less interactivity?”

- It's open
 - Full range of answers
- But it has too much context

All slides copyright 2000-2005, Mountain Goat Software

The best way to ask

“What would you be willing to give up in order to have it in a browser?”

- We want to ask questions that are
 - Open-ended
 - Context-free

All slides copyright 2000-2005, Mountain Goat Software

It's my problem, I know the solution

- Having a problem does not uniquely qualify you to solve it
- “It hurts when I go like this...”



All slides copyright 2000-2005, Mountain Goat Software

We need to stop asking users

- Since users don't know how to solve their problems, we need to stop *asking*
- We need to *involve* them instead

Empirical design

- Designers of the new system make decisions by studying prospective users in typical situations

Participatory design

- The users of the system become part of the team designing the behavior of the system

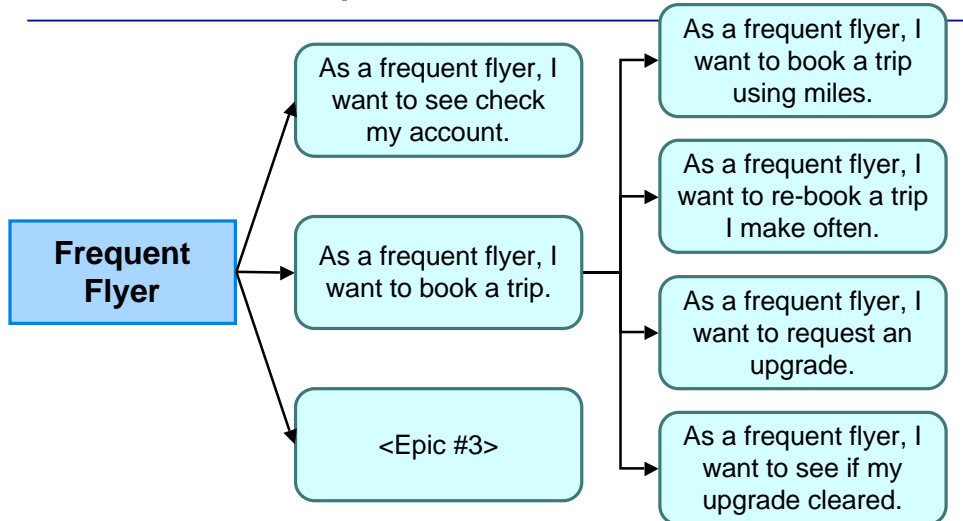
All slides copyright 2000-2005, Mountain Goat Software

Story-writing workshops

- Includes developers, users, customer, analysts, designers, everyone
- Goal is to brainstorm stories for the system
 - Some will be "implementation ready"
 - Others can be "epics"

All slides copyright 2000-2005, Mountain Goat Software

Start with epics and iterate



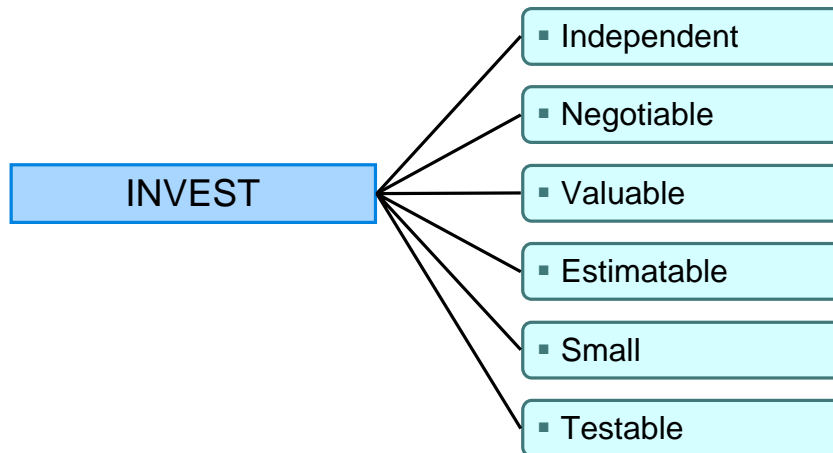
All slides copyright 2000-2005, Mountain Goat Software

Today's agenda

- What stories are
- User role modeling
- Story writing
- INVEST in good stories
- Why user stories?

All slides copyright 2000-2005, Mountain Goat Software

What makes a good story?



Thanks to Bill Wake for the acronym. See www.xp123.com.

All slides copyright 2000-2005, Mountain Goat Software

INVESTing in good stories

- **I**ndependent
 - Dependences lead to problems estimating and prioritizing
 - Can ideally select a story to work on without pulling in 18 other stories
- **N**egotiable
 - Stories are not contracts
 - Leave or imply some flexibility
- **V**aluable
 - To users or customers, not developers
 - Rewrite developer stories to reflect value to users or customers

All slides copyright 2000-2005, Mountain Goat Software

INVESTing in good stories

- **E**stimatable
 - Because plans are based on user stories, we need to be able to estimate them
- **S**mall
 - Complex stories are intrinsically large
 - Compound stories are multiple stories in one
- **T**estable
 - Stories need to be testable

All slides copyright 2000-2005, Mountain Goat Software

Small

- **L**arge stories (epics) are
 - hard to estimate
 - hard to plan
 - They don't fit well into single iterations
- **C**ompound story
 - An epic that comprises multiple shorter stories
- **C**omplex story
 - A story that is inherently large and cannot easily be disaggregated into constituent stories

All slides copyright 2000-2005, Mountain Goat Software

Compound stories

- Often hide a great number of assumptions

As a user, I can post my resume.

- A resume includes separate sections for education, prior jobs, salary history, publications, etc.
- Users can mark resumes as inactive
- Users can have multiple resumes
- Users can edit resumes
- Users can delete resumes

All slides copyright 2000-2005, Mountain Goat Software

Splitting a compound story

Split along operational boundaries (CRUD)

- As a user, I can create resumes, which include education, prior jobs, salary history, publications, presentations, community service, and an objective.
- As a user, I can edit a resume.
- As a user, I can delete a resume.
- As a user, I can have multiple resumes.
- As a user, I can activate and inactivate resumes.

All slides copyright 2000-2005, Mountain Goat Software

Splitting a compound story, cont.

Split along data boundaries

- As a user, I can add and edit educational information on a resume.
- As a user, I can add and edit prior jobs on a resume.
- As a user, I can add and edit salary history on a resume.
- As a user, I can delete a resume.
- As a user, I can have multiple resumes.
- As a user, I can activate and inactivate resumes.

All slides copyright 2000-2005, Mountain Goat Software

Other ways to split large stories

- Remove cross-cutting concerns
- Don't meet performance targets
- Avoid splitting stories into tasks

All slides copyright 2000-2005, Mountain Goat Software

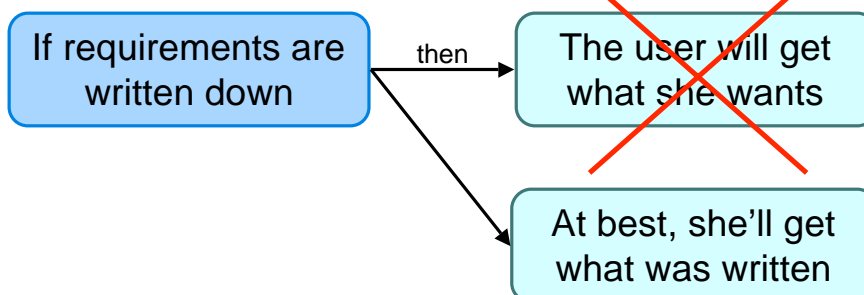
Today's agenda

- What stories are
- User role modeling
- Story writing
- INVEST in good stories
- Why user stories?

All slides copyright 2000-2005, Mountain Goat Software

So, why user stories?

- Shift focus from writing to talking



- "You built what I asked for, but it's not what I need."

All slides copyright 2000-2005, Mountain Goat Software

Words are imprecise

Meal comes with
soup or salad and
bread.

- (Soup or Salad) and Bread
- (Soup) or (Salad and Bread)

All slides copyright 2000-2005, Mountain Goat Software

Actual examples

The user can enter a
name. It can be 127
characters.

- Must the user enter a
name?
- Can it be other than 127
chars?

The system should
prominently display a
warning message
whenever the user enters
invalid data.

- What does *should* mean?
- What does *prominently
display* mean?
- Is *invalid data* defined
elsewhere?

All slides copyright 2000-2005, Mountain Goat Software

Additional reasons

- Stories are comprehensible
 - Developers and customers understand them
 - People are better able to remember events if they are organized into stories[†]
- Support and encourage iterative development
 - Can easily start with epics and disaggregate closer to development time

[†]Bower, Black, and Turner. 1979.
Scripts in Memory for Text.

All slides copyright 2000-2005, Mountain Goat Software

Yet more reasons

- Stories are the right size for planning
- Stories support opportunistic development
 - We design solutions by moving opportunistically between top-down and bottom-up approaches[†]
- Stories support participatory design

[†]Guindon. 1990. *Designing the Design Process.*

All slides copyright 2000-2005, Mountain Goat Software

User stories show the big picture

- All requirements are not equal
 - “Designers fix a top-level concept based on their initial understanding of a problem.”[†]
 - “May produce a solution for only the first few requirements they encounter.”[‡]

Sources: [†]*Making Use* by John M. Carroll (2000) and
[‡]*Technology and Change* by D.A. Schon (1967).

All slides copyright 2000-2005, Mountain Goat Software

What are we building?

IEEE 830 Software Requirements Spec

6. The product shall have a gas engine.
7. The product shall have four wheels.
 1. The product shall have a rubber tire mounted to each wheel.
8. The product shall have a steering wheel.
9. The product shall have a steel body.

Source: Adapted from *The Inmates are Running the Asylum* by Alan Cooper (1999).

All slides copyright 2000-2005, Mountain Goat Software

What if we had stories instead?

As a user, I want to mow my lawn quickly and easily.

As a user, I want to be comfortable while mowing my lawn.

All slides copyright 2000-2005, Mountain Goat Software

The product



All slides copyright 2000-2005, Mountain Goat Software

Most importantly...

Don't forget the purpose

- The story text we write on cards is less important than the conversations we have.
- "Stories represent requirements, they do not document them."[†]

[†]Rachel Davies, "The Power of Stories," XP 2001.

All slides copyright 2000-2005, Mountain Goat Software

Mike Cohn contact information



- mike@mountaingoatsoftware.com
- (303) 810-2190 (mobile)
- (720) 890-6110 (office)

▪ www.mountaingoatsoftware.com



All slides copyright 2000-2005, Mountain Goat Software

Additional guidelines for good stories

- Start with goals
- Slice the cake
- Write closed stories
- Put constraints on cards
- Size the story to the horizon
- Keep the UI out as long as possible
- Some things aren't stories
- Include user roles in the stories
- Write for one user
- Write in active voice

Don't forget the purpose

All slides copyright 2000-2005, Mountain Goat Software

Stories are not use cases

Title: Accept reservation for a room.

Primary Actor: Purchaser

...

Main Success Scenario:

1. Purchaser submits credit card number, date, and authentication information.
2. System validates credit card.
3. System charges credit card full amount for all nights of stay.
4. Purchaser is given a unique confirmation number.

All slides copyright 2000-2005, Mountain Goat Software

Stories are not use cases

Extensions:

2a The card is not a type accepted by the system.

2a1 System notifies the user to use a different card.

2b The card is expired.

2b1 System notifies the user to use a different card.

3a The card has insufficient available credit.

3a1 System charges as much as it can to the current card.

3b1 User is told about the problem and asked to enter a second card; use case continues at 2

All slides copyright 2000-2005, Mountain Goat Software

Differences between use cases / stories

- Scope
- Completeness
- Longevity
- Purpose
 - Use cases
 - Document agreement between customer and developers
 - Stories
 - Written to facilitate release and iteration planning
 - Placeholders for future conversations

All slides copyright 2000-2005, Mountain Goat Software