

Effective User Stories for Agile Requirements

Mike Cohn

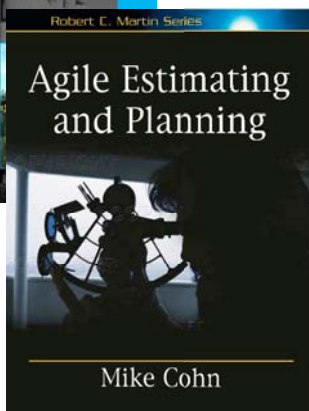
Mountain Goat Software

mike@mouнтаingoatsoftware.com



1

Mike Cohn - background



Consultant, author,
and speaker

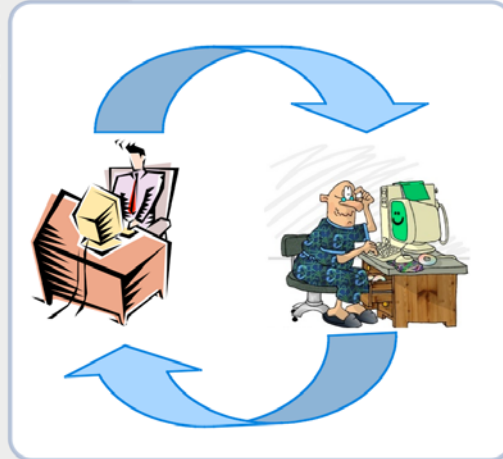
- Founding member and director of Agile Alliance, Scrum Alliance, and Agile Project Leadership Network
- Founder of Mountain Goat Software



2

It's a communication problem

- Software requirements is a *communication problem*
- Those who want software must communicate with those who will build it



Copyright Mountain Goat Software, LLC

3

Balance is critical

- If either side dominates, the business loses
- If the business side dominates...
 - ...functionality and dates are mandated with little regard for reality or whether the developers understand the requirements
- If the developers dominate...
 - ...technical jargon replaces the language of the business and developers lose the opportunity to learn from listening



Copyright Mountain Goat Software, LLC

4

Resource allocation

- We need a way of working together so that resource allocation becomes a shared problem
- Project fails when the problem of resource allocation falls too far to one side



Copyright Mountain Goat Software, LLC

5

Responsibility for resource allocation

If developers shoulder the responsibility...

- May trade quality for additional features
- May only partially implement a feature
- May solely make decisions that should involve the business side

If the business shoulders the responsibility...

- Lengthy upfront requirements negotiation and signoff
- Features are progressively dropped as the deadline nears



Copyright Mountain Goat Software, LLC

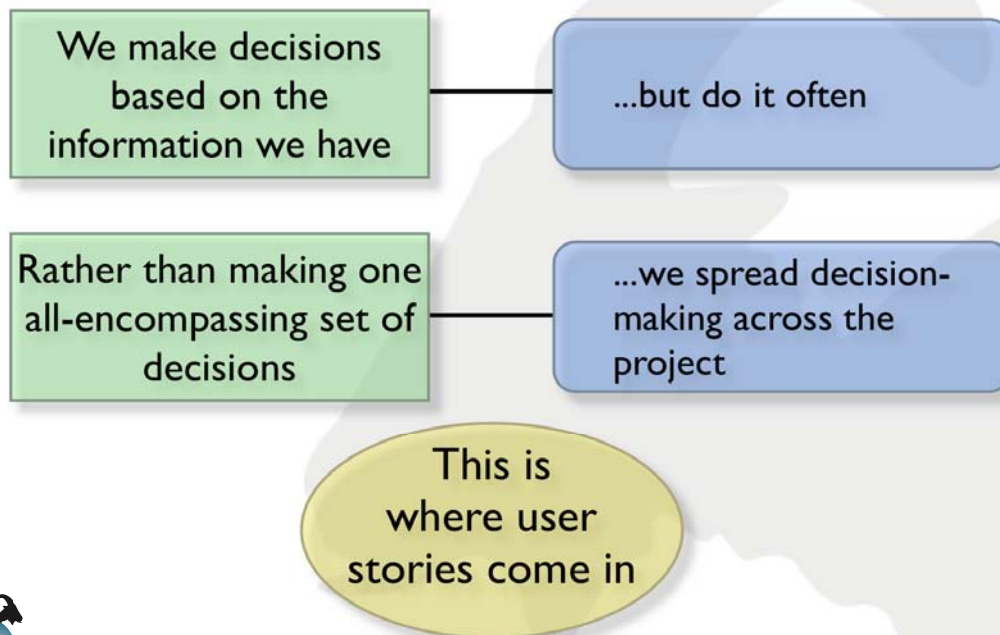
6

Imperfect schedules

- We cannot perfectly predict a software schedule
 - As users see the software, they come up with new ideas
 - Too many intangibles
 - Developers have a notoriously hard time estimating
- If we can't perfectly predict a schedule, we can't perfectly say what will be delivered



So what do we do?



Today's agenda



- What stories are
- User role modeling
- Story writing
- INVEST in good stories
- What user stories are not
- Why user stories



Copyright Mountain Goat Software, LLC

9

Ron Jeffries' Three Cs

Card

- Stories are traditionally written on note cards.
- Cards may be annotated with estimates, notes, etc.

Conversation

- Details behind the story come out during conversations with product owner

Confirmation

- Acceptance tests confirm the story was coded correctly



Source: XP Magazine 8/30/01, Ron Jeffries.

Copyright Mountain Goat Software, LLC

10

Samples from a travel website

As a user, I want to reserve a hotel room.

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

As a user, I want to cancel a reservation.

As a frequent flyer, I want to rebook a past trip, so that I save time booking trips I take

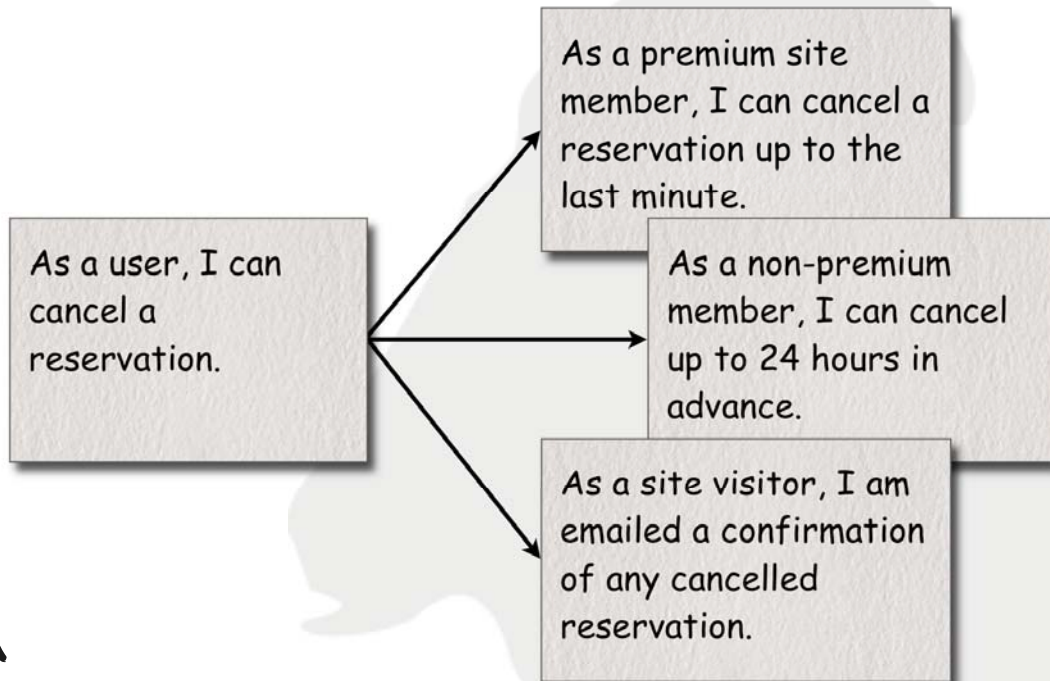


Where are the details?

- 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?



Details added in smaller sub-stories

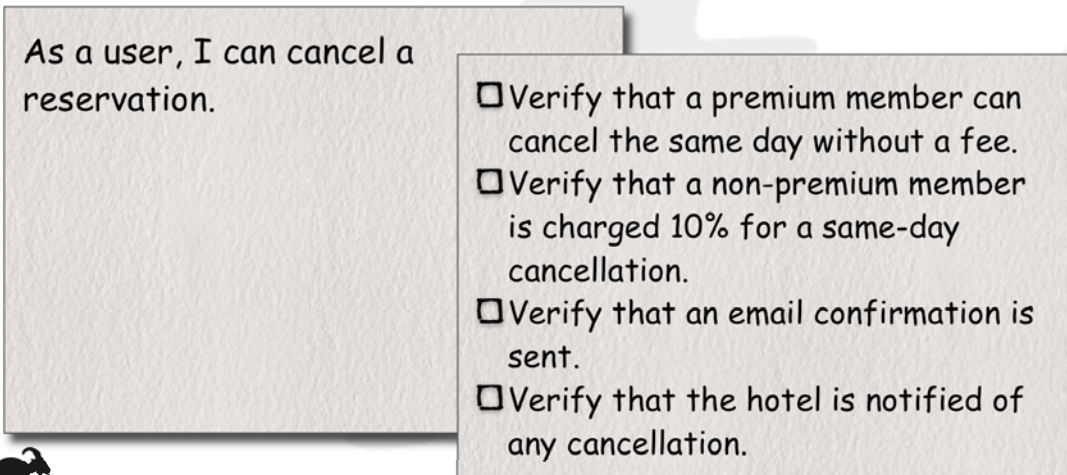


Copyright Mountain Goat Software, LLC

13

Details as conditions of satisfaction

- The product owner's conditions of satisfaction can be added to a story
 - These are essentially tests



Copyright Mountain Goat Software, LLC

14

An example

As a VP Marketing, I want to review the performance of historical promotional campaigns so that I can identify and repeat profitable ones.

An epic;
weeks to implement

As a VP Marketing, I want to select the timeframe to use when reviewing the performance of past promotional campaigns, so that I can identify and repeat profitable ones.

Implementation-size stories;
days to implement

As a VP Marketing, I can select which type of campaigns (direct mail, TV, email, radio, etc.) to include when reviewing the performance of historical promotional campaigns.



Copyright Mountain Goat Software, LLC

15

An example

As a VP Marketing, I want to see information on direct mailings when reviewing historical campaigns.

As a VP Marketing, I want to see information on television advertising when reviewing historical campaigns.

As a VP Marketing, I want to see information on email advertising when reviewing historical campaigns.



Copyright Mountain Goat Software, LLC

16

Today's agenda



- What stories are
- User role modeling
- Story writing
- INVEST in good stories
- What user stories are not
- Why user stories



“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



Travel Site—Who's the user?

Frequent flier who never knows where she'll be

Wants to schedule her family's annual vacation

Frequent flier who flies every week but always to the same place

A frequent flier's assistant; books her reservations

Hotel chain Vice President; wants to monitor reservations



Copyright Mountain Goat Software, LLC

19

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

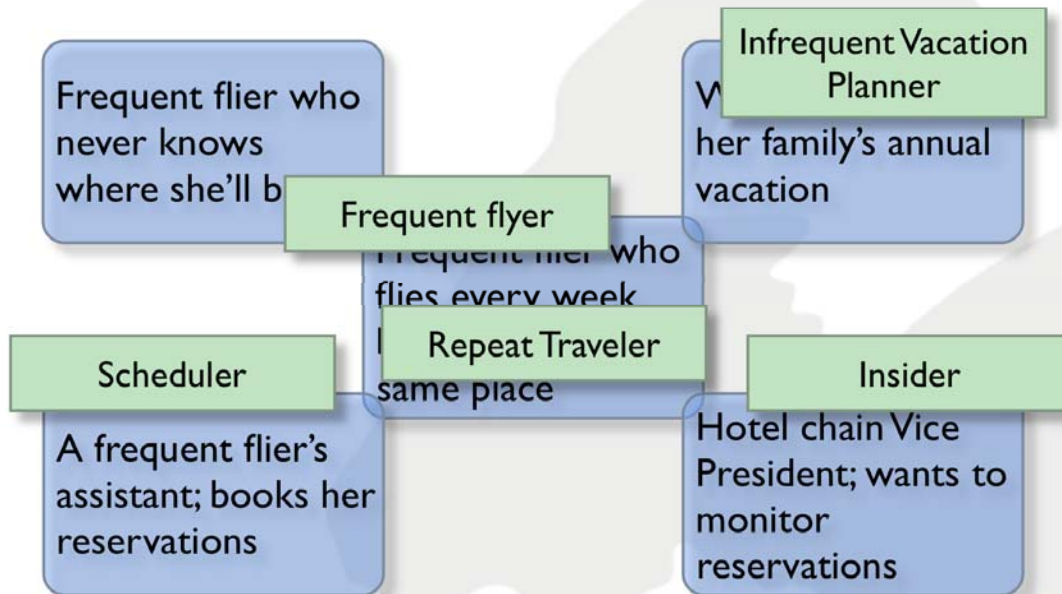


Source: *Software for Use* by Constantine and Lockwood (1999).

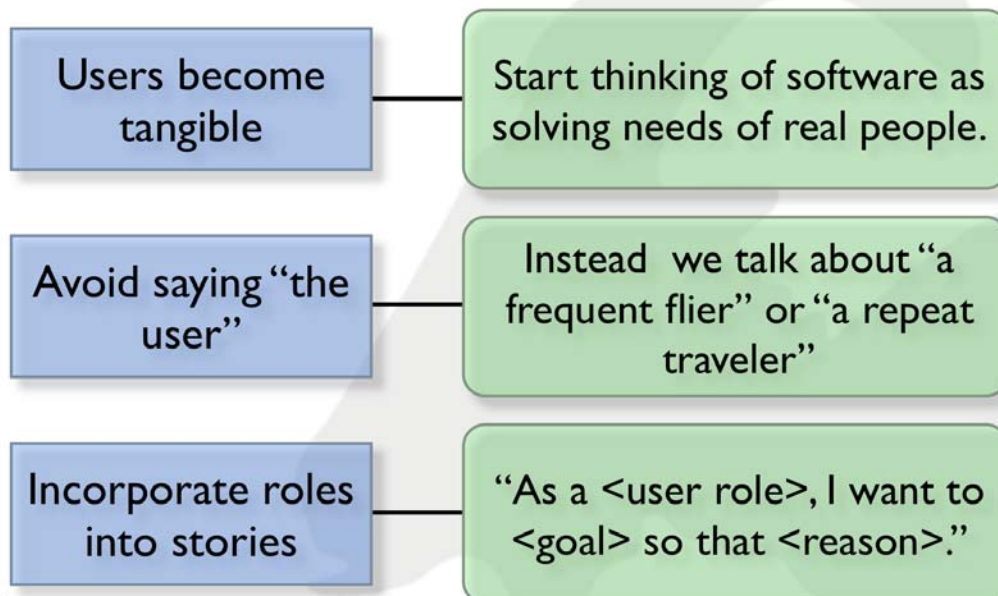
Copyright Mountain Goat Software, LLC

20

Common attributes



Advantages of using roles



Today's agenda



- What stories are
- User role modeling
- Story writing
- INVEST in good stories
- What user stories are not
- Why user stories



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}



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



Copyright Mountain Goat Software, LLC

25

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



Copyright Mountain Goat Software, LLC

26

Context matters

“My wife and I split up...”



“He’s no longer with us...”



Copyright Mountain Goat Software, LLC

27

My context isn't your context

“Dad, make it warmer.”



“Increase the temperature.”

“Move the temperature closer to what we call warm.”



Copyright Mountain Goat Software, LLC

28

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...”



Copyright Mountain Goat Software, LLC

29

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



Copyright Mountain Goat Software, LLC

30

Story-writing workshops

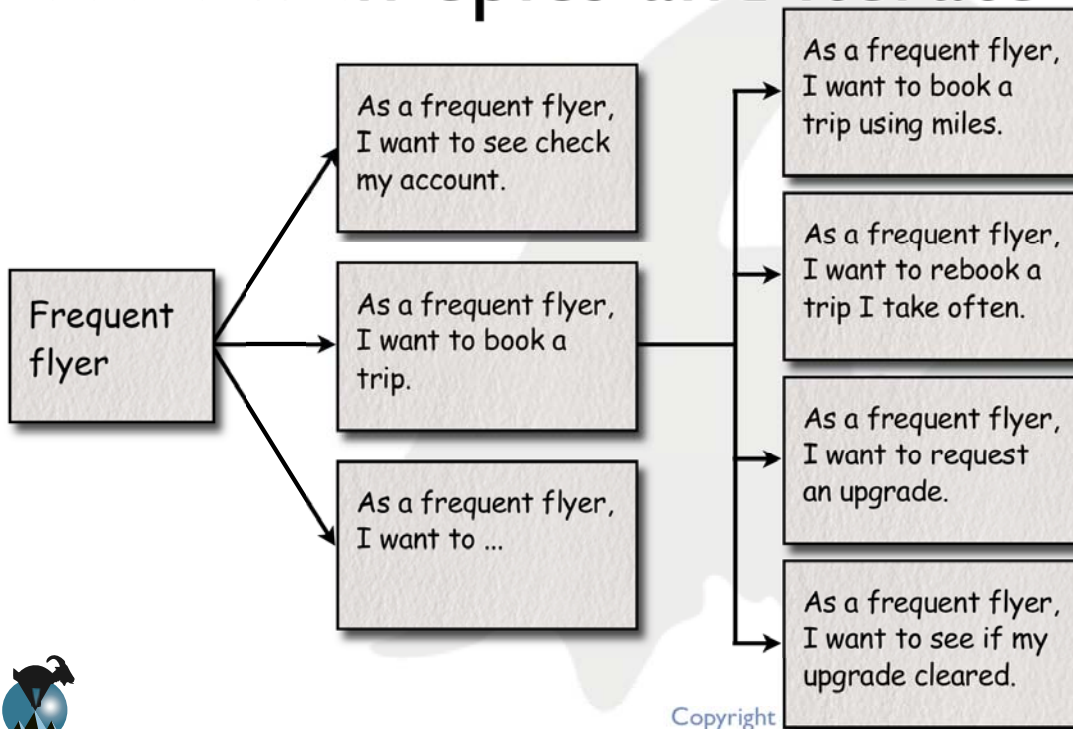
- Includes developers, users, customer, others
- Brainstorm to generate stories
- Goal is to write as many stories as possible
 - Some will be “implementation ready”
 - Others will be “epics”
- No prioritization at this point



Copyright Mountain Goat Software, LLC

31

Start with epics and iterate



Copyright

32

Today's agenda



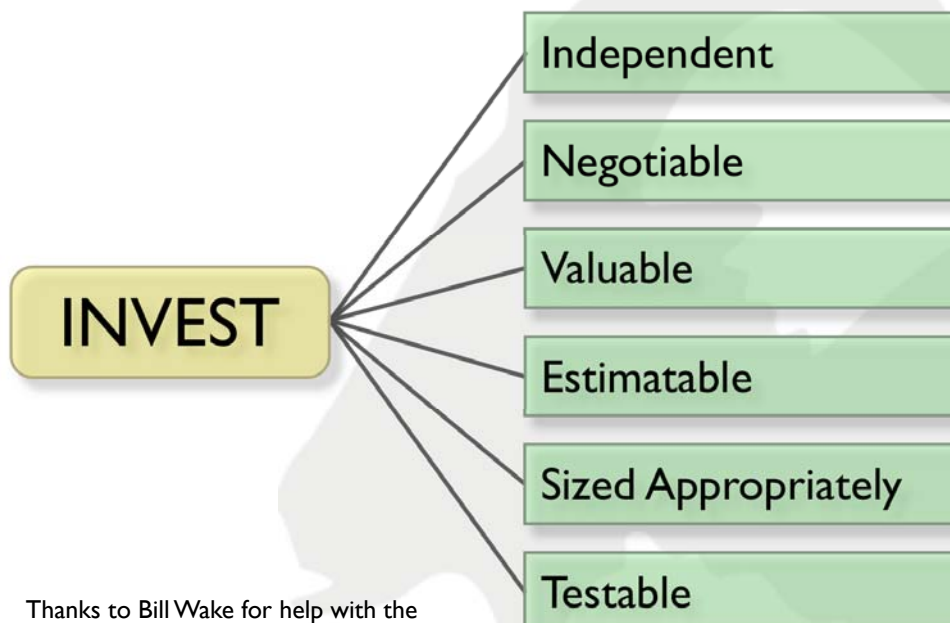
- What stories are
- User role modeling
- Story writing
- INVEST in good stories
- What user stories are not
- Why user stories



Copyright Mountain Goat Software, LLC

33

What makes a good story?



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



Copyright Mountain Goat Software, LLC

34

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



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



Today's agenda



- What stories are
- User role modeling
- Story writing
- INVEST in good stories
- What user stories are not
- Why user stories

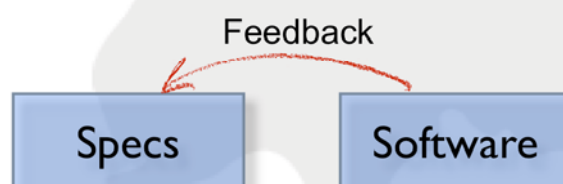


Copyright Mountain Goat Software, LLC

37

Software requirements specs

- Problems with IEEE 830-style Software Requirements Specifications
 - Tedious to read
 - So readers skim or skip sections
 - Assumes everything is knowable in advance



Copyright Mountain Goat Software, LLC

38

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).



Copyright Mountain Goat Software, LLC

39

What are we building?

IEEE 830 Software Requirements Spec

1. The product shall have a gas engine.
2. The product shall have four wheels.
 - 2.1. The product shall have a rubber tire mounted to each wheel.
3. The product shall have a steering wheel.
4. The product shall have a steel body.

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



Copyright Mountain Goat Software, LLC

40

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.



Copyright Mountain Goat Software, LLC

41

The product



Copyright Mountain Goat Software, LLC

42

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.
- 5.



Copyright Mountain Goat Software, LLC

43

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



Copyright Mountain Goat Software, LLC

44

Differences: use cases / stories

- Scope
- Completeness
- Longevity
- Purpose
 - Use cases document an agreement between the customer and the developers
 - User stories are reminders to have a conversations and are written to facilitate planning



Today's agenda

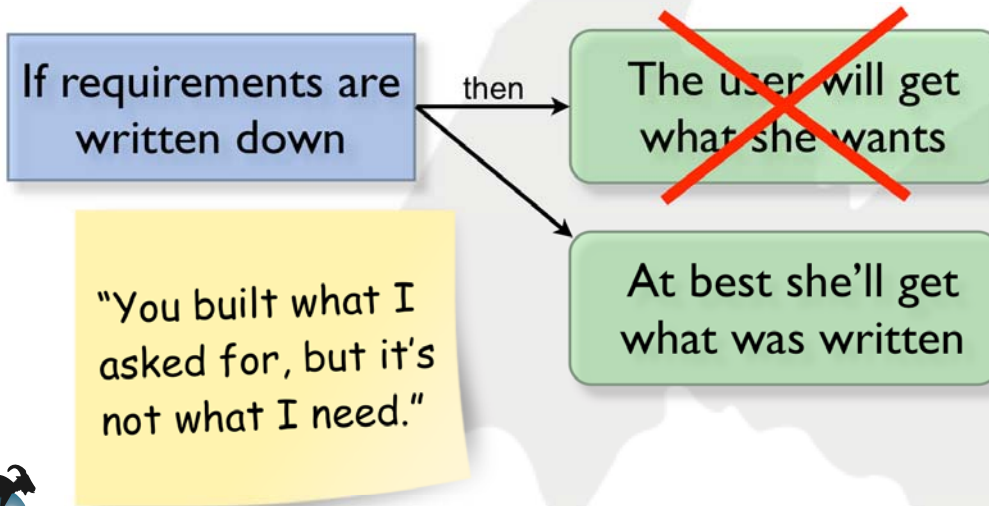


- What stories are
- User role modeling
- Story writing
- INVEST in good stories
- What user stories are not
- Why user stories



So why user stories?

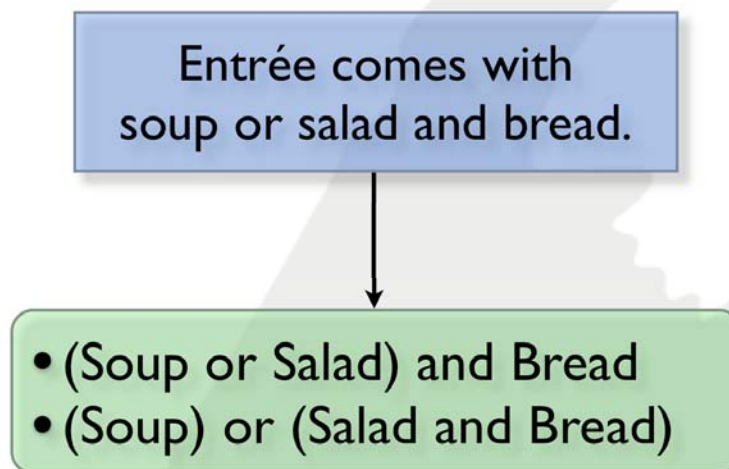
- Shift focus from writing to talking



Copyright Mountain Goat Software, LLC

47

Words are imprecise



Copyright Mountain Goat Software, LLC

48

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?



Another example

“I handed in a script last year and the studio didn’t change one word.”

“The word they didn’t change was on page 87.”

~Steve Martin



Additional reasons

- Stories are comprehensible
 - Developers and customers understand them
 - People are better able to remember events if they are organized into stories[†]
- Stories are the right size for planning
- 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*.

Copyright Mountain Goat Software, LLC

51

Yet more reasons

- 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*.

Copyright Mountain Goat Software, LLC

52

Why not user stories?

- There are some drawbacks to be aware of
 - On a large project there can be lots of stories
 - Solution is to remember to keep some as epics initially
 - Or “staple” some together into themes
 - Stories on cards don’t facilitate traceability
 - But you can do it, or you can use software
 - Stories focus on increasing tacit, not formal, communication
 - May need to supplement with some formal documentation



Most importantly...

Don't forget the purpose

The story text we write on cards is less important than the conversations we have.



Upcoming public classes

Date	What	Where
September 26-27	Certified ScrumMaster	London
September 28	Agile Estimating & Planning	London
November 7-8	Certified ScrumMaster	Santa Clara
November 9	Agile Estimating & Planning	Santa Clara
November 29-30	Certified Product Owner (with Ken Schwaber)	Boulder
January 16-17	Certified ScrumMaster (with Ken Schwaber)	Orlando

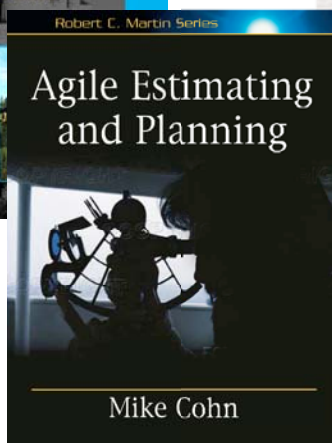
Register at
www.mountaingoatsoftware.com

Copyright Mountain Goat Software, LLC

55



Mike Cohn contact info



mike@mountaingoatsoftware.com

www.mountaingoatsoftware.com

(720) 890-6110 (office)

(303) 810-2190 (mobile)



Copyright Mountain Goat Software, LLC

56

