

Effective User Stories for Agile Requirements

Mike Cohn
Mountain Goat Software
mike@moungaigoatsoftware.com

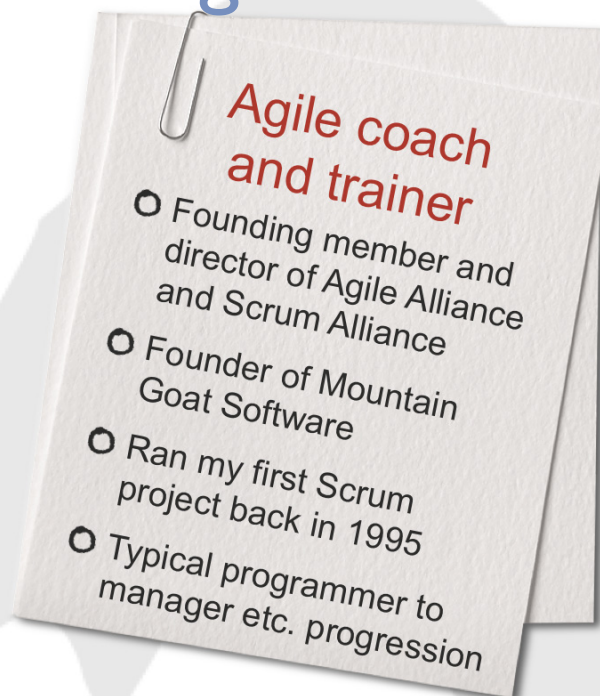
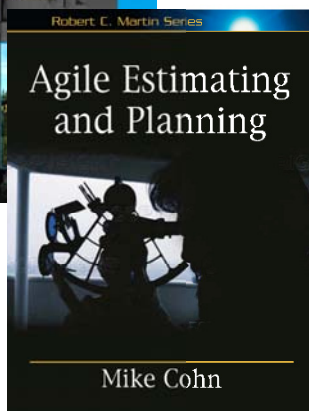


March 10, 2009

© 2003-2009 Mountain Goat Software®

1

Mike Cohn - background

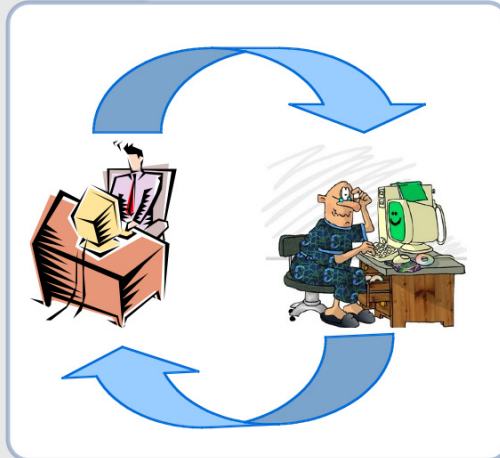


© 2003-2009 Mountain Goat Software®

2

What problem do stories address?

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



© 2003–2009 Mountain Goat Software®

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



© 2003–2009 Mountain Goat Software®

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



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



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?

We make decisions based on the information we have

...but do it often

Rather than making one all-encompassing set of decisions

...we spread decision-making across the project

This is where user stories come in



Agenda

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



Poor requirements

Poor requirements are often listed as one of the chief causes of project failure.

1. What is a bad requirements process?
2. What are some problems that result from a bad requirements process?





What Stories Are



© 2003–2009 Mountain Goat Software®

11

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.

© 2003–2009 Mountain Goat Software®

12

Samples from a travel website

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

As a vacation traveler, 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 often.



A recommendation

Use this template

“As a <user role>, I want <goal> so that <reason>.”



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

As a user, I can cancel a reservation.

As a premium site member, I can cancel a reservation up to the last minute

As a non-premium member, I can cancel up to 24 hours in advance.

As a site visitor, I am emailed a confirmation of any cancelled reservation.



Details as conditions of satisfaction

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

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.



A game development example

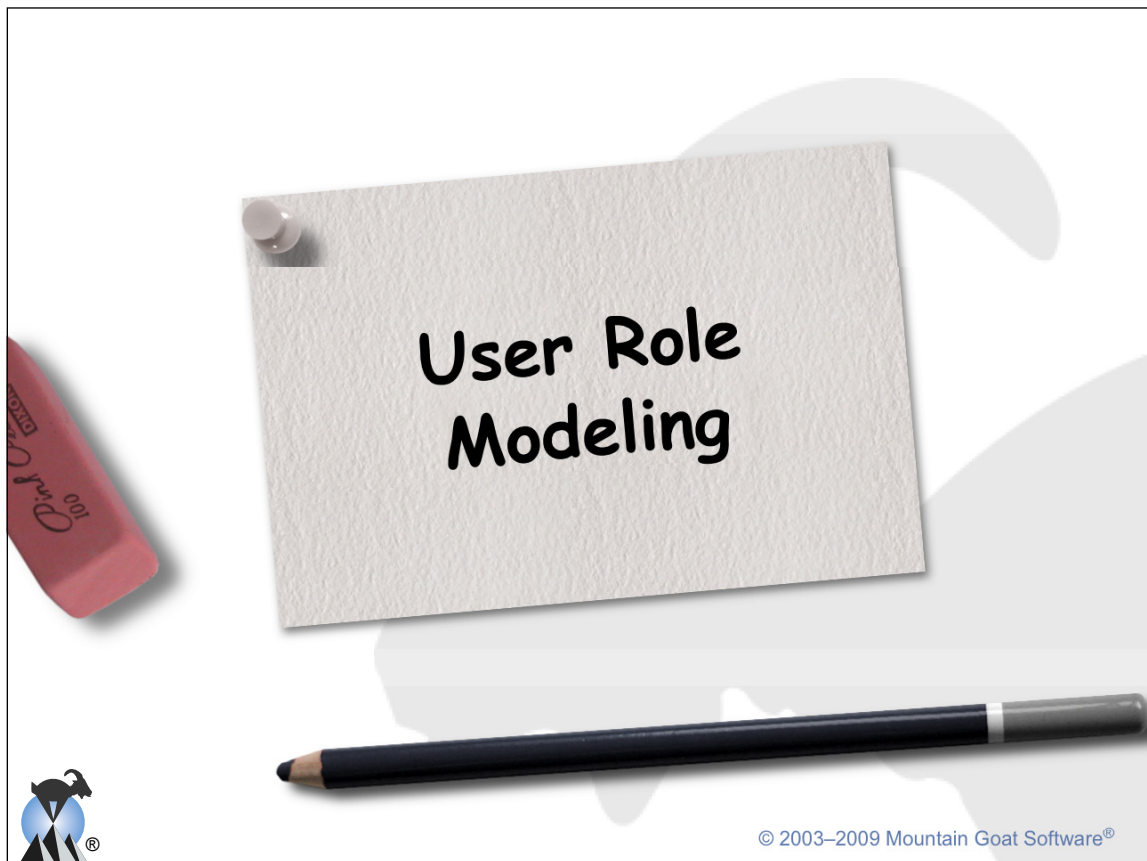
As a player, I want enemies to have physical reactions when I hit them.

- An enemy twists left when hit on the left and right when hit on the right.
- The enemy staggers back when hit in the center.
- The enemy tilts back when hit in the head.



Techniques can be combined

- These approaches are not mutually exclusive
- Write stories at an appropriate level
- By the time it's implemented, each story will have conditions of satisfaction associated with it



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

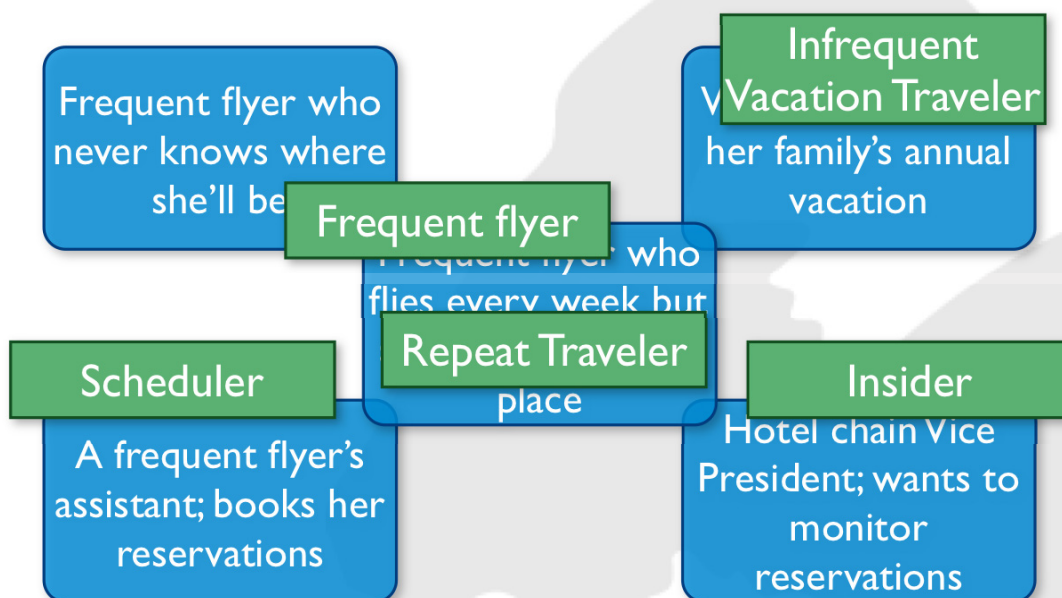


Source: *Software for Use* by Constantine and Lockwood (1999).

© 2003–2009 Mountain Goat Software®

21

Common attributes



© 2003–2009 Mountain Goat Software®

22

Habbo Hotel user roles

- Achievers
- Traditionals
- Creatives
- Rebels
- Loners



© 2003–2009 Mountain Goat Software®

23

High net worth user roles

- Family steward
 - Taking care of family money. Goals like children's tuition and passing on wealth. Open to estate planning. Typically conservative.
- Phobic
 - Don't like investing and don't understand it.
- Independent
 - Wants the freedom of financial independence.
- Anonymous
 - Extremely private, don't want to disclose any information. Very loyal after initial trust is gained.



© 2003–2009 Mountain Goat Software®

24

More high net worth roles

- Mogul
 - Seeks power, influence and control. Investing is another way to show their importance.
- VIP
 - Want prestige. Want investments to help buy possessions.
- Accumulator
 - Live below their means and don't show outward signs of affluence.
- Gambler
 - Invests for the excitement, drama, and performance results.
- Innovator
 - Like new products, strategies, services. Often from a technical background.

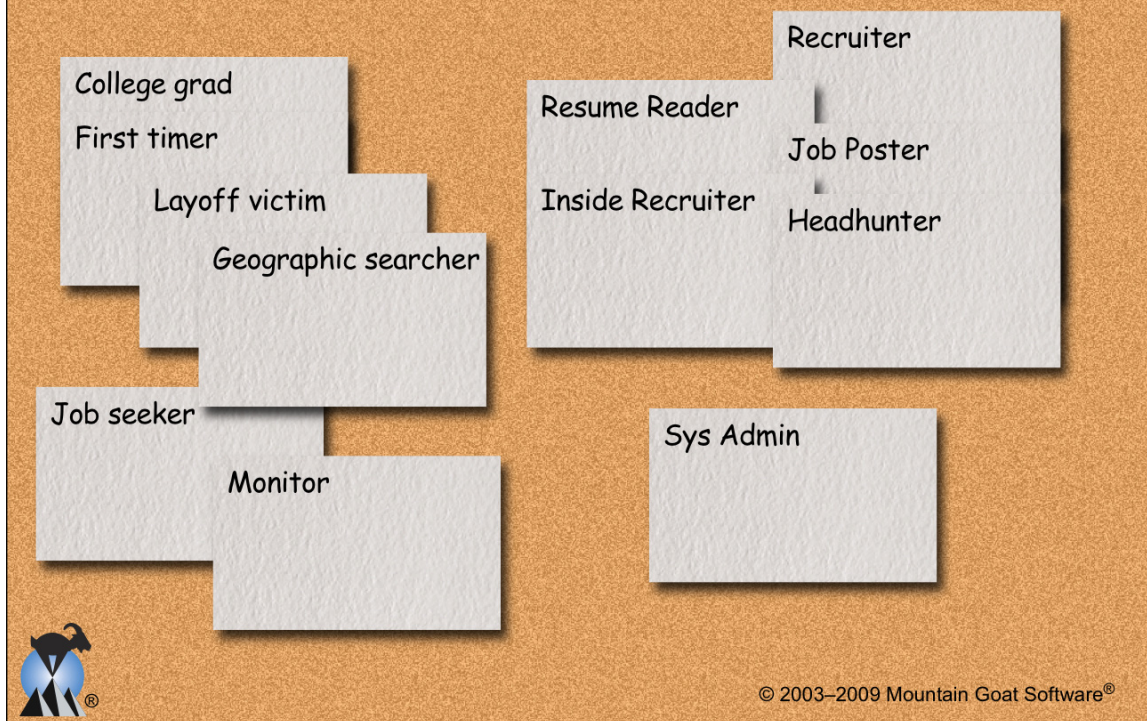


User role brainstorming


- Brainstorming meeting
 - Customer, developers, anyone who understands a product's intended users
- Everyone grabs a stack of cards
- Write role names on cards
 - As fast as possible and with no judgment
 - No turns
 - Place card on table
 - Call out role name as you place it



Organize, consolidate, refine



College grad
First timer
Layoff victim
Geographic searcher
Job seeker
Monitor
Resume Reader
Inside Recruiter
Recruiter
Job Poster
Headhunter
Sys Admin

 © 2003–2009 Mountain Goat Software®

27

User role brainstorming



We've been hired by CondoXChange to develop a website with these features:

- Condo owners can rent their condos
- Swap for condos in other cities
- Find service providers (cleaners, etc.)
- Advertising for various services

1. Brainstorm the user roles who will interact with this site.
2. Organize, consolidate and refine them into a usable user role model.



28

Advantages of using roles

Avoid saying “the user”

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

Users become tangible

Start thinking of software as solving needs of real people.

Incorporate roles into stories

“As a <user role>, I want to <goal> so that <benefit>.”



System and programmer users

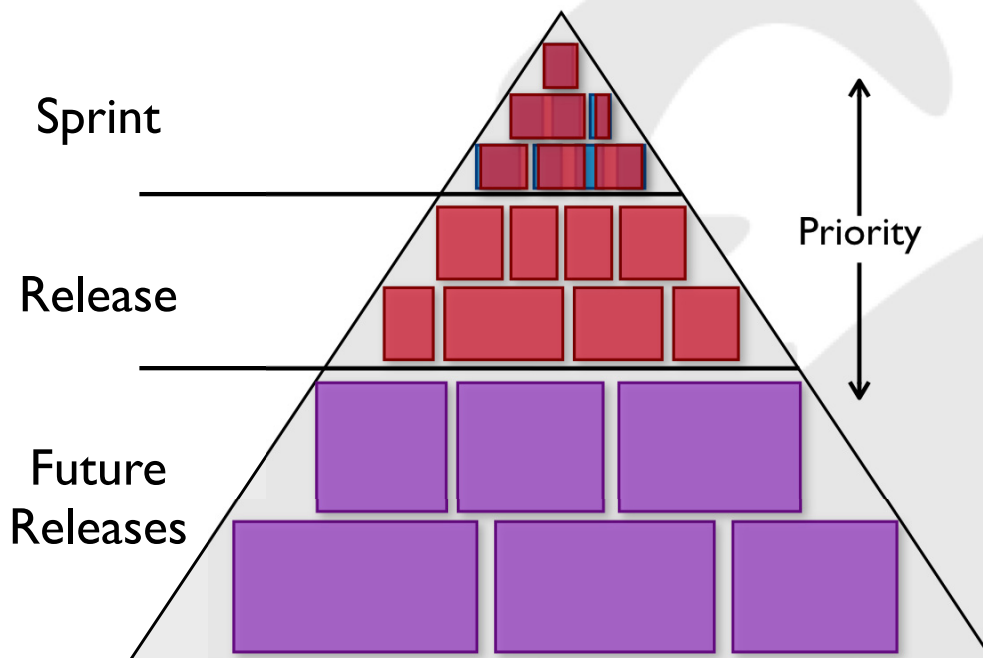
As the payment verification system, I want all transactions to be well-formed XML.

As a programmer, I want an API for deleting widgets from the database.

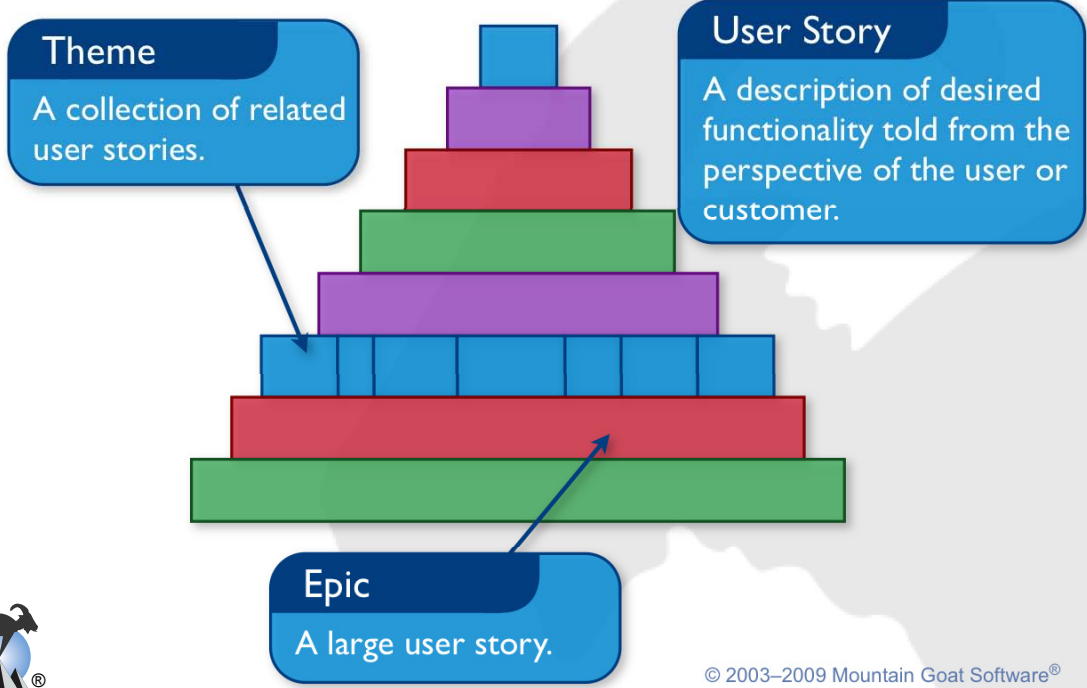




The product backlog iceberg



Stories, themes and epics



33

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.

Clearly an epic

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.

Epics??

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.

© 2003–2009 Mountain Goat Software®

34

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.



© 2003–2009 Mountain Goat Software®

35

Finally, add the Conditions Of Satisfaction

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

- See how many viewers by age range.
- See how many viewers by income level.



© 2003–2009 Mountain Goat Software®

36

Another example

As a player I want online multiplayer so I can connect to the internet and play against other players online.

Clearly an epic

As a player I want to have a player lobby so I can see the online games available for me to join and be able to join one.

Epics??



© 2003–2009 Mountain Goat Software®

37

An example

As a player I want to see how many players are in a game in progress.

As a player I want to join a game in progress.

As a player I want to start a new game.



© 2003–2009 Mountain Goat Software®

38

Story-writing workshops

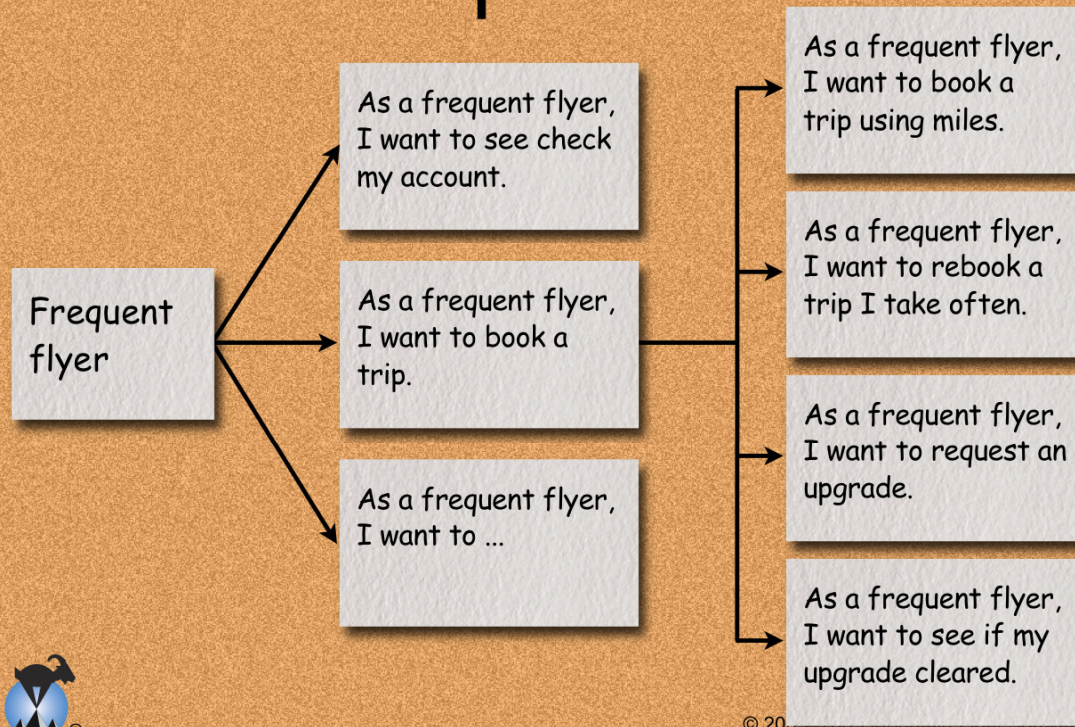
- Includes developers, users, customer, others
- Brainstorm to generate stories
- Goal is to write as many stories as possible
 - Some will be “implementation ready”
 - Others will be “epics”
- No prioritization at this point



© 2003–2009 Mountain Goat Software®

39

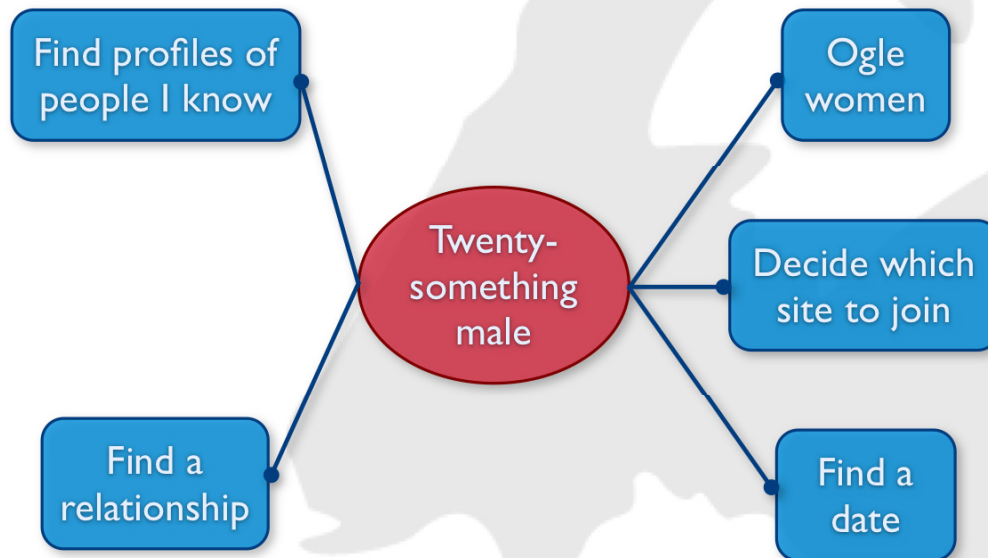
Start with epics and iterate



© 2003–2009 Mountain Goat Software®

40

Or do a mindmap



© 2003–2009 Mountain Goat Software®

41

Another approach

- Walk through a low-fidelity (paper) user interface
 - Ask open-ended, context-free questions as you go:
 - What will the users most likely want to do next?
 - What mistakes could the user make here?
 - What could confuse the user at this point?
 - What additional information could the user need?
 - Consider these questions for each user role



© 2003–2009 Mountain Goat Software®

42

A story-writing workshop



CondoXChange

With the user roles you identified earlier, conduct a story-writing workshop for CondoXChange. Identify top-level goal stories / epics for these roles. Then convert a couple of epics into more implementation-size stories.

Use this template

“As a <user role>, I want <goal> so that <reason>.”



© 2003–2009 Mountain Goat Software®

43

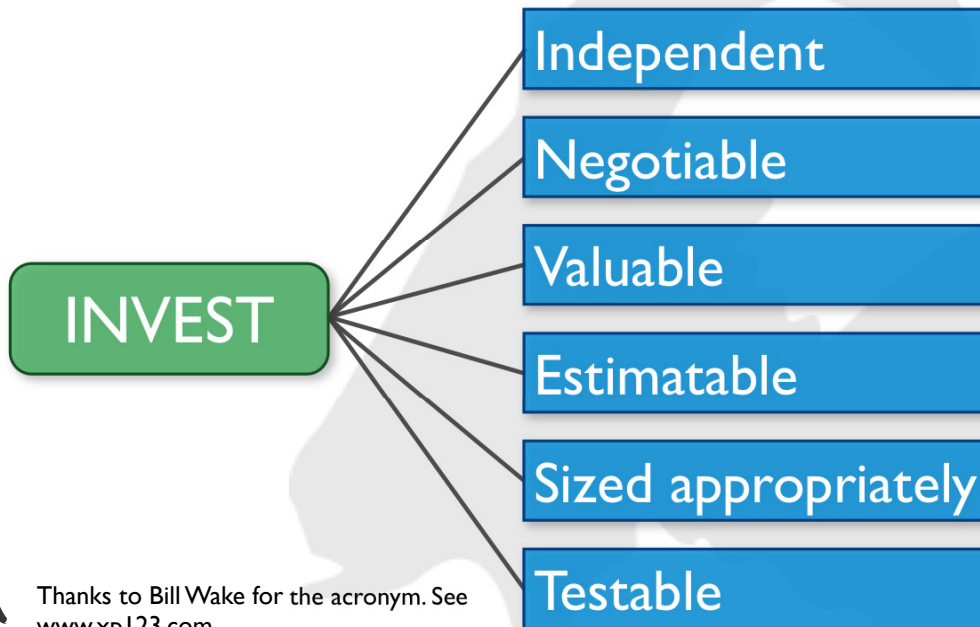
INVEST
in
Good Stories



© 2003–2009 Mountain Goat Software®

44

What makes a good story?



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

© 2003–2009 Mountain Goat Software®

45

Independent

- Avoid introducing dependencies
 - Leads to difficulty prioritizing and planning

As a customer, I can pay for the items in my cart with a Visa card.

As a customer, I can pay for the items in my cart with a MasterCard.

As a customer, I can pay for the items in my cart with an American Express card.

- First story will take 3 days to develop
- It doesn't matter which is first
- Others will each take 1 day



© 2003–2009 Mountain Goat Software®

46

Making stories independent

Combine the stories

- As a customer, I can pay with a credit card.

Split across a different dimension

- As a customer, I can pay with a first type of credit card.
- As a customer, I can pay two additional types of credit card.

Write two estimates and move on

- 3 days if done first; 1 otherwise



What about this approach?

Extract technical commonalities

- As a programmer, I need to code the infrastructure for processing credit cards.
- As a customer, I can pay with a Visa.
- As a customer, I can pay with a MasterCard,
- As a customer, I can pay with an American Express.

- Sometimes necessary but not ideal
- Why?



Negotiable

- Stories are not contracts
 - They do not need to include all details
- Leave some flexibility in some stories to be worked out during the iteration



Which is more negotiable?

1

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.



2

As a user, I can add printers to the printer list.

- Auto-search
- Manually specify DNS name
- Manually specify IP address

Note: I've got some "advanced" ways to add printers, too. See me if you have time.



Valuable

- Stories must be valuable to either:

Users

- As a user, I can search for a job by title and salary range.

Customers

- As the sponsor of this project, I want it to pass an ISO 9001 audit.
- As a sponsor of this project, I want it to produce documentation in compliance with CMMI level 3.
- As a system administrator, I want all configuration information for all users stored in a central location.



Stories valued by developers

- Should be rewritten to show the benefit to users or customers

All connections to the database are through a connection pool.

As a purchaser of this system, I want it usable by 50 users with a five-user database license.



Rewrite this as a story:

Refactor the payroll processing code.

As a
I want
so that

Refactor

To change the structure but not the behavior of code.



Estimatable

- Because stories are used in planning
- A story may not be estimatable if

Developers lack domain knowledge

As a new user, I am given a diabetic screening.

Developers lack technical knowledge

As any user, I can zoom in on a map without delay.

The story is too big

As a job seeker, I can find a job.

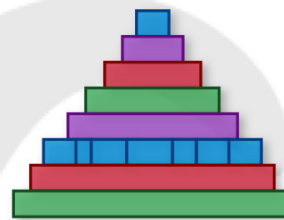


© 2003–2009 Mountain Goat Software®

55

Sized appropriately

- Small stories for the near future
- Epics for further out
- Stories are progressively refined as the time to do them moves closer
- Two types of large stories
 - Complex stories: Inherently large and cannot be made smaller
 - Compound stories: Multiple stories combined into one



© 2003–2009 Mountain Goat Software®

56

Compound stories

- An epic that comprises multiple shorter stories
- Often hide a great number of assumptions

As a seller [on an auction site], I can post items for sale.

- To post an item for sale you must provide multiple data elements (description, auction end date, etc.)
- Some data elements are required, some are optional.
- Items can be updated after posted.
- Auctions can be cancelled.



Splitting a compound story

Split along operational boundaries (CRUD)

- As a seller, I can create a new auction listing.
- As a seller, I can update an existing auction listing.
- As a seller, I can delete an auction listing.



Splitting a compound story

Split along data boundaries

- As a seller, I can create and update the description of an auction item.
- As a seller, I can add, update or remove a photo from an auction listing.



© 2003–2009 Mountain Goat Software®

59

Testable

- Tests demonstrate that a story meets the customer's expectations
- Automate, automate, automate

A user must find the software easy to use.

As a novice user, I am able to complete common workflows without training.


A user must never have to wait long for a screen to appear.

As a user, I want to see new screens within 2 seconds 95% of the time.



© 2003–2009 Mountain Goat Software®

60



What User Stories Are Not



© 2003–2009 Mountain Goat Software®

61

User stories are not...

- IEEE 830 Software Requirements Specifications
 - “The system shall...”
- Use Cases

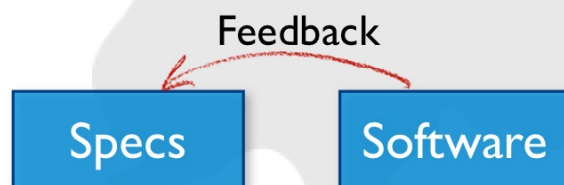


© 2003–2009 Mountain Goat Software®

62

Problems with IEEE 830

- Time-consuming to write and read
- Tedious to read
 - So readers skim or skip sections
- Assumes everything is knowable in advance



© 2003–2009 Mountain Goat Software®

63

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).



© 2003–2009 Mountain Goat Software®

64

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: *The Inmates Are Running the Asylum*
by Alan Cooper, 1999.

© 2003–2009 Mountain Goat Software®

65

What if we had stories instead?

66

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.



© 2003–2009 Mountain Goat Software®

67

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.



© 2003–2009 Mountain Goat Software®

68

Differences: 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



© 2003–2009 Mountain Goat Software®

69



Why User Stories



© 2003–2009 Mountain Goat Software®

70

So why user stories?

- Shift focus from writing to talking

If requirements are written down

then

~~The user will get what she wants~~

At best she'll get what was written

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



Words are imprecise

Entrée comes with soup or salad and bread.

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



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 understandable
 - 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*.



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*.



© 2003–2009 Mountain Goat Software®

75

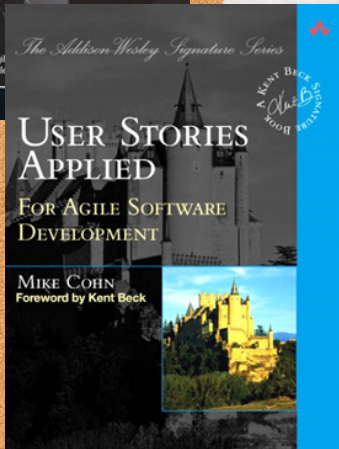
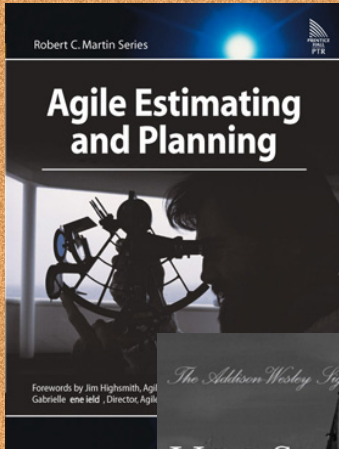
Date	What	Where
Mar 31–Apr 1 Apr 2	Certified ScrumMaster Agile Estimating and Planning	Seattle
May 12 May 13–14 May 15	Effective User Stories Certified ScrumMaster Agile Estimating and Planning	Orlando
June 8 June 9–10 June 11	Effective User Stories Certified ScrumMaster Agile Estimating and Planning	San Jose
August 3 August 4–5 August 6	Effective User Stories Certified ScrumMaster Agile Estimating and Planning	Denver
Sept 14–15 Sept 16–17	Certified ScrumMaster Certified Scrum Product Owner	La Jolla

Information and registration at
www.mountaingoatsoftware.com



© 2003–2009 Mountain Goat Software®

76



Mike Cohn
mike@mountaingoatsoftware.com
www.mountaingoatsoftware.com
(720) 890-6110



MOUNTAIN GOAT
SOFTWARE



© 2003–2009 Mountain Goat Software®