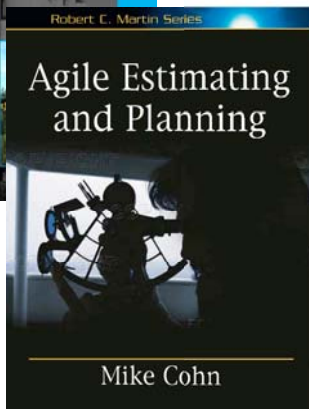# User Stories for Agile Requirements

1

Mike Cohn - background

Consultant, author, and speaker

○ Founding member and director of Agile Alliance, Scrum Alliance, and Agile Project Leadership Network

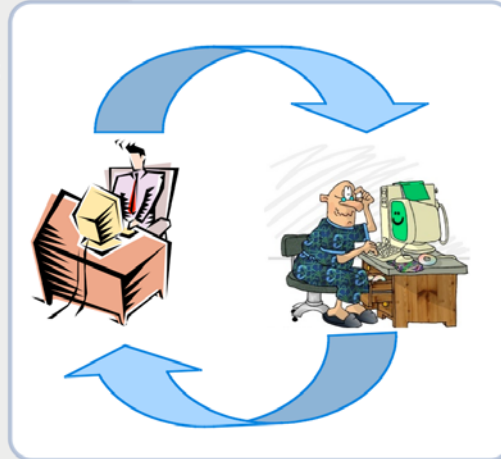○ Founder of Mountain Goat Software

Copyright Mountain Goat Software, LLC

2

It's a communication problem

- Software requirements is a *communication problem*

- Those who want software must communicate with those who will build it

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

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

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
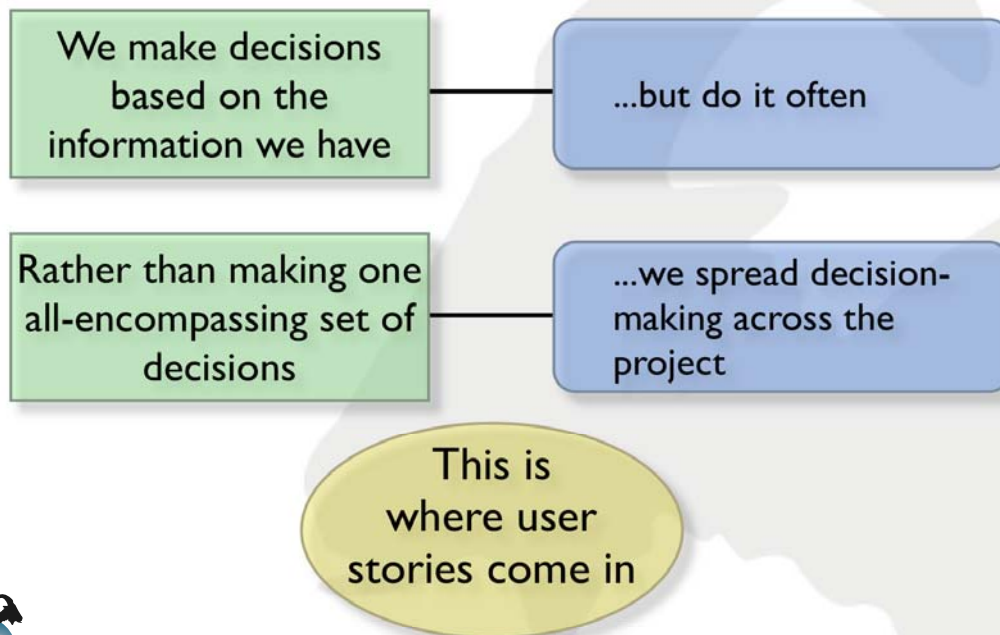- Features are progressively dropped as the deadline nears

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?

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

Today's agenda

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

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.

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
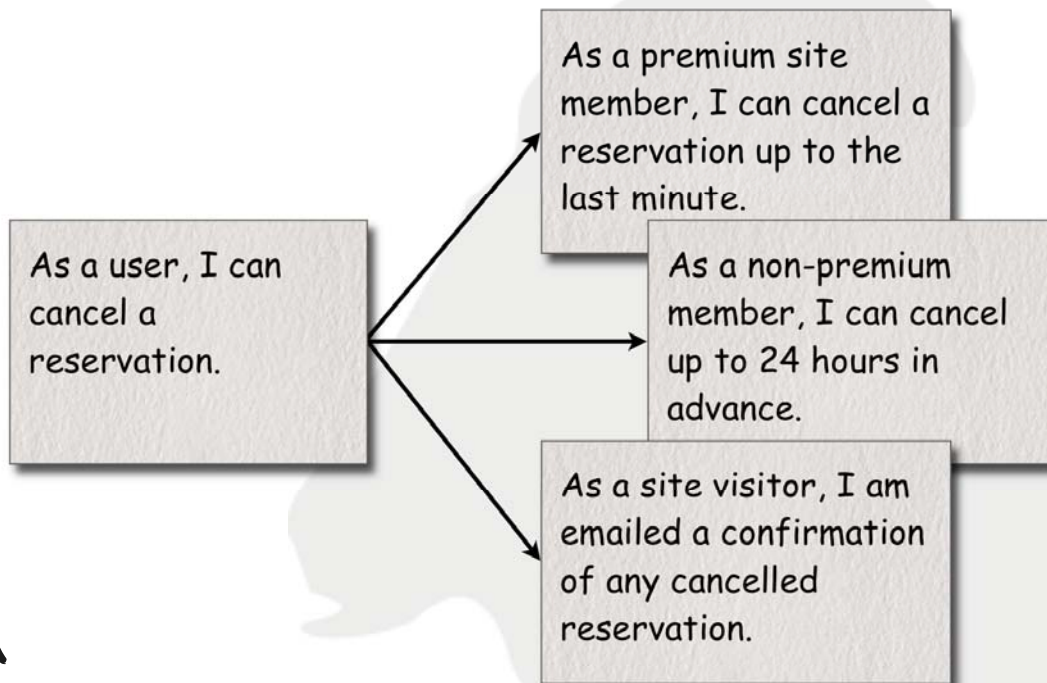
11

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?
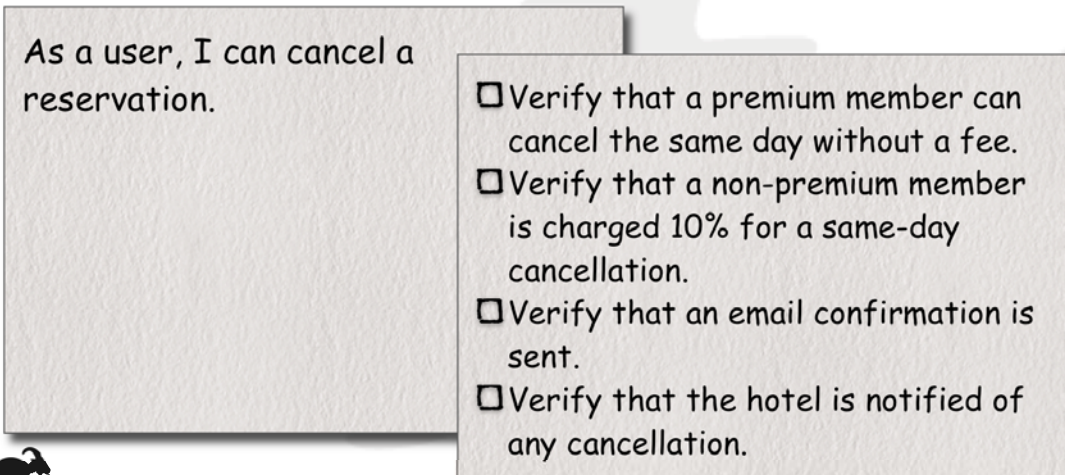
12

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.

13

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.

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.

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.

Today's agenda

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

17

"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

18

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

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

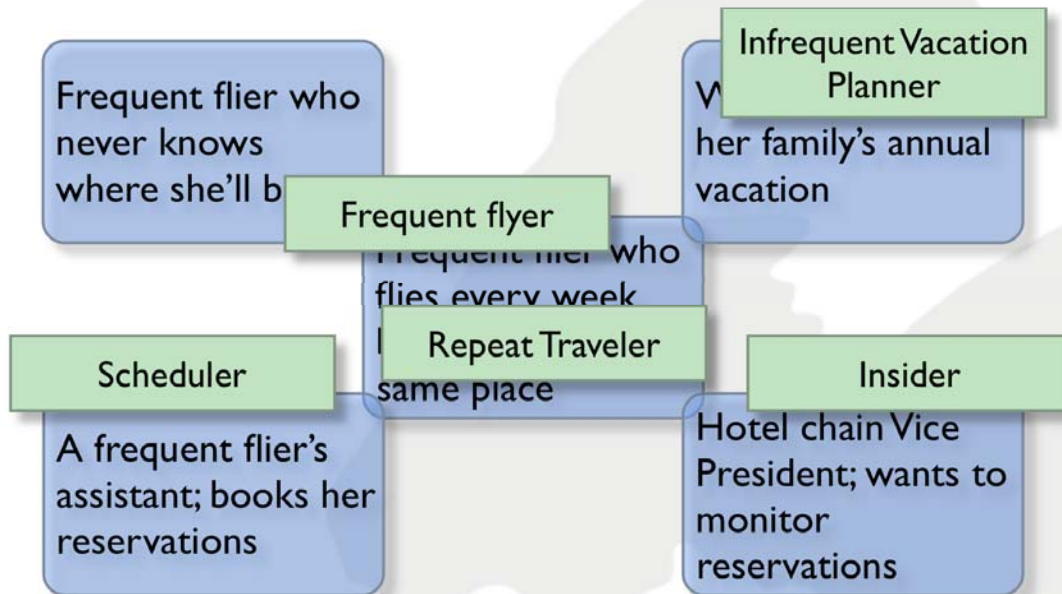- Used extensively in usage-centered design

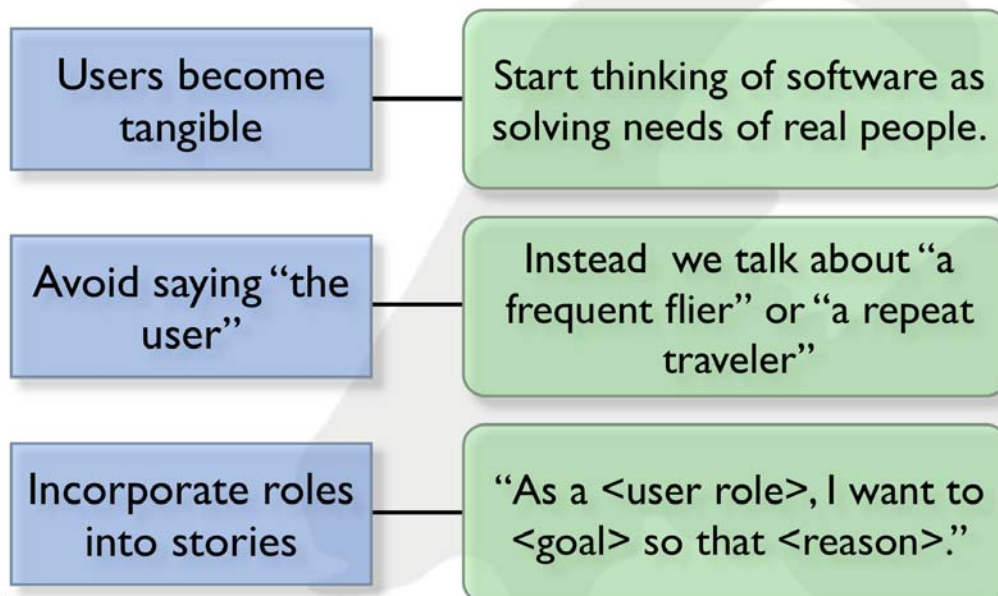Source: *Software for Use* by Constantine and Lockwood (1999).

20

Common attributes

Frequent flier who never knows where she'll b...

Frequent flyer

~~Frequent flier who flies every week...~~
...same place

Infrequent Vacation Planner

W... her family's annual vacation

Repeat Traveler

Scheduler

A frequent flier's assistant; books her reservations

Insider

Hotel chain Vice President; wants to monitor reservations

Copyright Mountain Goat Software, LLC

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 <user role>, I want to <goal> so that <reason>."

Copyright Mountain Goat Software, LLC

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

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

Context matters

"My wife and I split up…"

"He's no longer with us…"

27

My context isn't your context

"Dad, make it warmer."

"Increase the temperature."

"Move the temperature closer to what we call warm."

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…"

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

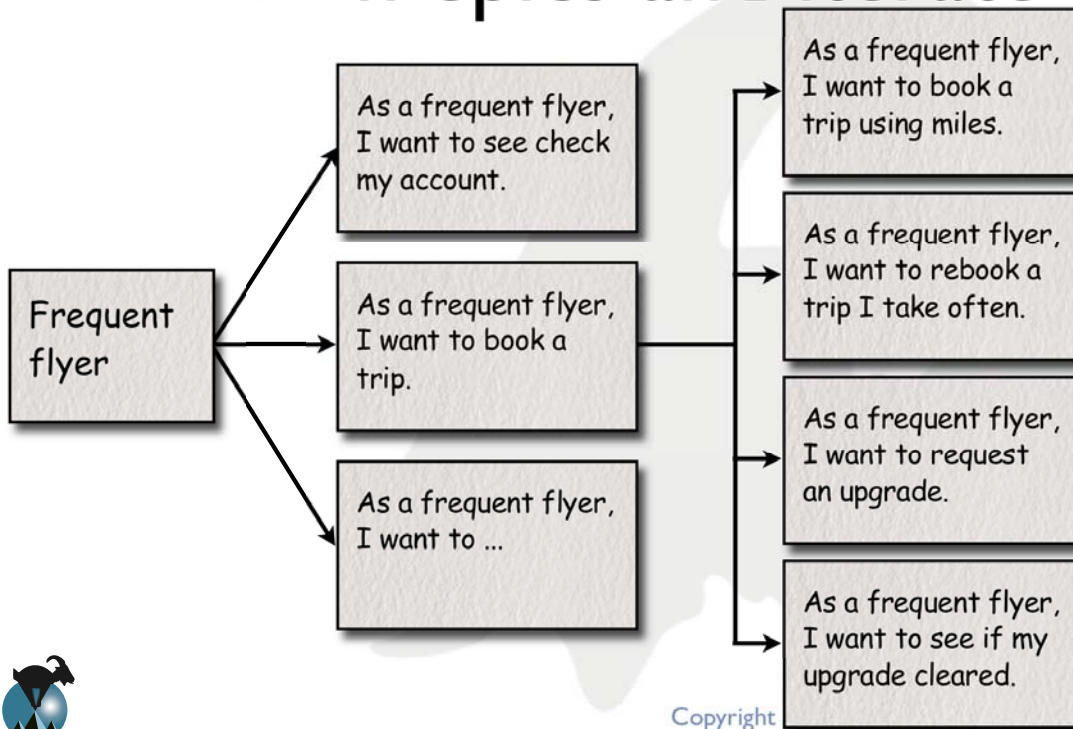| Empirical design	• Designers of the new system make decisions by studying prospective users in typical situations
Participatory design	• The users of the system become part of the team designing the behavior of the system

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
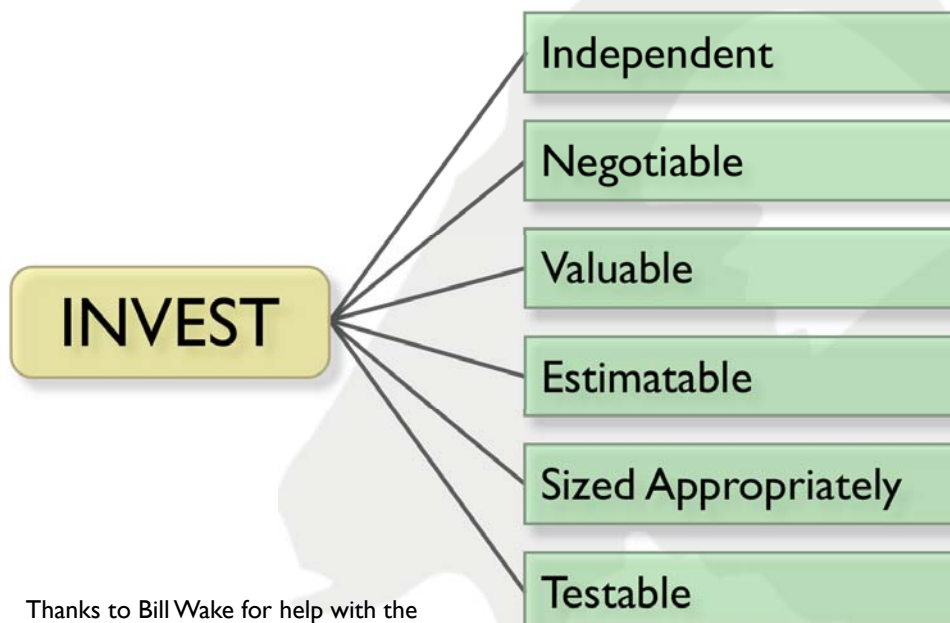
31

Start with epics and iterate

As a frequent flyer, I want to see check my account.

Frequent flyer

As a frequent flyer, I want to book a trip.

As a frequent flyer, I want to ...

As a frequent flyer, I want to book a trip using miles.

As a frequent flyer, I want to rebook a trip I take often.

As a frequent flyer, I want to request an upgrade.

As a frequent flyer, I want to see if my upgrade cleared.

32

Today's agenda

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

What makes a good story?

INVEST

- Independent
- Negotiable
- Valuable
- Estimatable
- Sized Appropriately
- Testable

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

INVESTing in good stories

- **Independent**
  - Dependenices lead to problems estimating and prioritizing
  - Can ideally select a story to work on without pulling in 18 other stories
- **Negotiable**
  - Stories are not contracts
  - Leave or imply some flexibility
- **Valuable**
  - To users or customers, not developers
  - Rewrite developer stories to reflect value to users or customers

INVESTing in good stories

- **Estimatable**
  - Because plans are based on user stories, we need to be able to estimate them
- **Sized Appropriately**
  - Complex stories are intrinsically large
  - Compound stories are multiple stories in one
- **Testable**
  - 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

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

Feedback

Specs   Software

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).

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).

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.

41

The product

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.

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

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
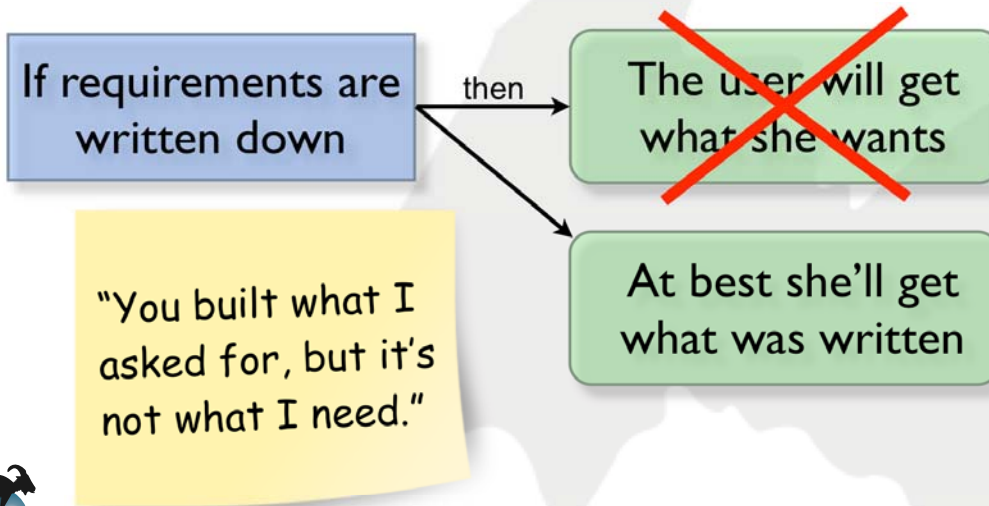
Today's agenda

☑ What stories are

☑ User role modeling

☑ Story writing

☑ INVEST in good stories

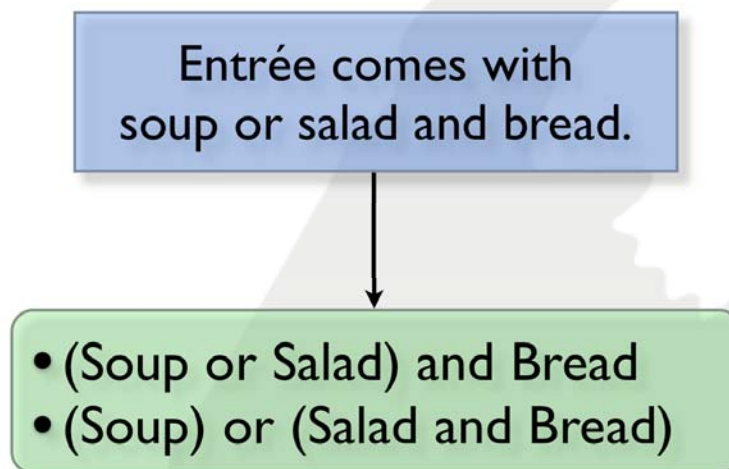☑ What user stories are not

☐ Why user stories

So why user stories?

- Shift focus from writing to talking

If requirements are written down → then → The user will get what she wants ~~(crossed out)~~

↓

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)

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[†]

- 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.*

51

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.

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

53

Most importantly…

Don't forget the purpose

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

54

Upcoming public classes

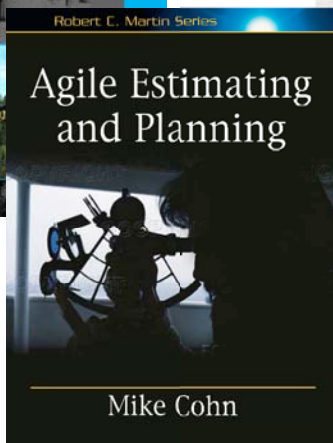| Date	What	Where
August 1-2	Certified ScrumMaster	Denver
August 3	Agile Estimating & Planning	Denver
November 7-8	Certified ScrumMaster	Santa Clara
November 9	Agile Estimating & Planning	Santa Clara
November 29-30	Certified Product Owner (with Ken Schwaber)	Boulder

Register at
www.mountaingoatsoftware.com

Mike Cohn contact info

mike@mountaingoatsoftware.com

www.mountaingoatsoftware.com

(720) 890-6110 (office)

(303) 810-2190 (mobile)

MOUNTAIN GOAT
S O F T W A R E