User Stories for Agile Requirements

1

Mike Cohn - background

USER STORIES APPLIED
For Agile Software Development
MIKE COHN
Foreword by Kent Beck

Robert C. Martin Series
Agile Estimating and Planning
Mike Cohn
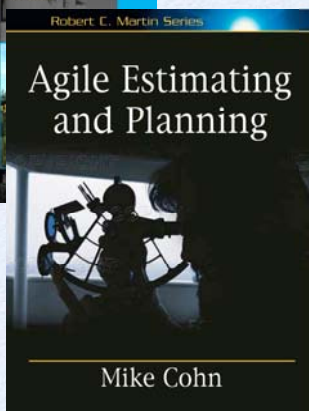
Consultant, author, and speaker

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

2

- Software requirements is a *communication problem*

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

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

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

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
- ☐ Story writing
- ☐ INVEST in good stories
- ☐ Why user stories

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.

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 flier, 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

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

Today's agenda

- ☑ What stories are
- ☐ Story writing
- ☐ INVEST in good stories
- ☐ Why user stories

15

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}

16

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

17

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

18

Context matters

"My wife and I split up…"

"He's no longer with us…"

"Dad, make it warmer."

19

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…"

20

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

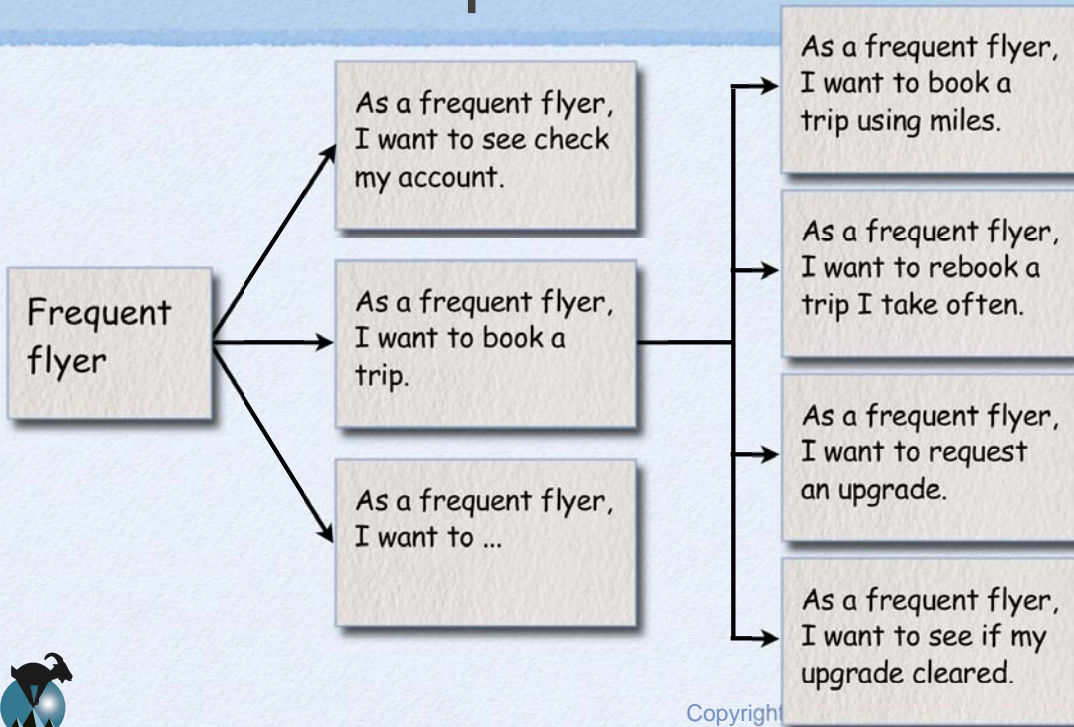| Empirical design	• Designers of the new system make decisions by studying prospective users in typical situations
Participatory design	• The users of the system become part of the team designing the behavior of the system

21

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

22

Start with epics and iterate

Frequent flyer

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

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.

23

A template I like

As a ‹type of user›, I want to ‹goal› so that ‹reason›.

24
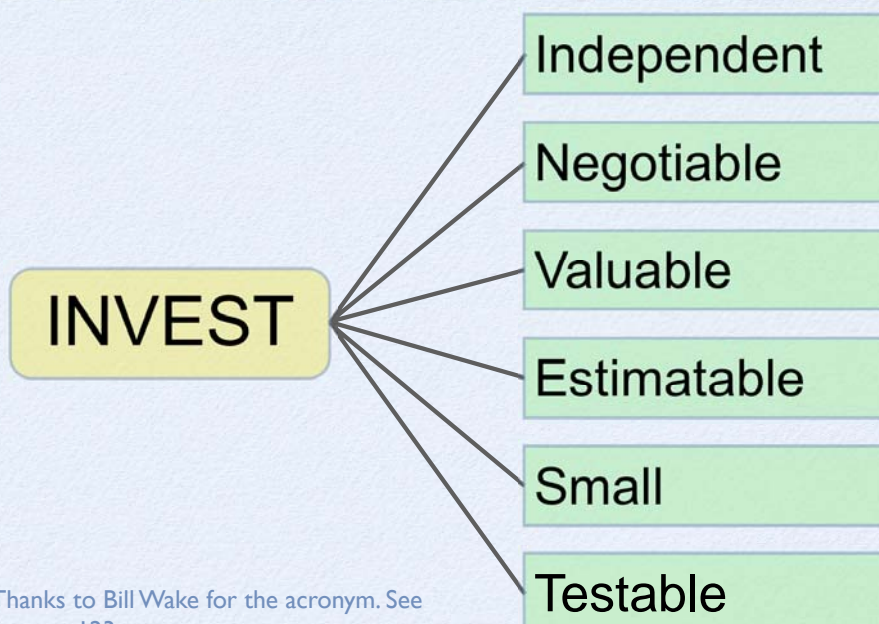
Today's agenda

☑ What stories are

☑ Story writing

☐ INVEST in good stories

☐ Why user stories

What makes a good story?

INVEST

- Independent
- Negotiable
- Valuable
- Estimatable
- Small
- Testable

Thanks to Bill Wake for 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

27

INVESTing in good stories

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

28

Small

- Large stories (epics) are

  - Hard to estimate

  - Hard to plan

    - Won't fit in a single iteration

  - Two types of large story

    - Complex story

    - Compound story

Compound stories

- An epic that comprises multiple shorter stories

- Often hide a great number of assumptions

As a job seeker, I can post my resume.

→

- A resume includes multiple sections (education, job history, salary history, publications, etc.).
- Resumes can be inactive or active.
- A user can have multiple resumes.
- A user can edit a resume.
- A user can delete her resume.

Splitting a compound story

Split along operational boundaries (CRUD)

- As a job seeker, I can create a new resume.
- As a job seeker, I can edit a resume.
- As a job seeker, I can delete a resume.
- As a job seeker, I can have multiple resumes.
- As a job seeker, I can activate and inactivate resumes.

oat Software, LLC

Splitting a compound story

Split along data boundaries

- As a job seeker, I can add and update educational information.
- As a job seeker, I can add and update my job history.
- As a job seeker, I can add and edit salary history on a resume.
- As a job seeker, I can delete a resume.
- As a job seeker, I can have multiple resumes.
- As a job seeker, I can activate and inactivate resumes.

at Software, LLC

Today's agenda

- ☑ What stories are
- ☑ Story writing
- ☑ INVEST in good stories
- ☐ Why user stories

33

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."
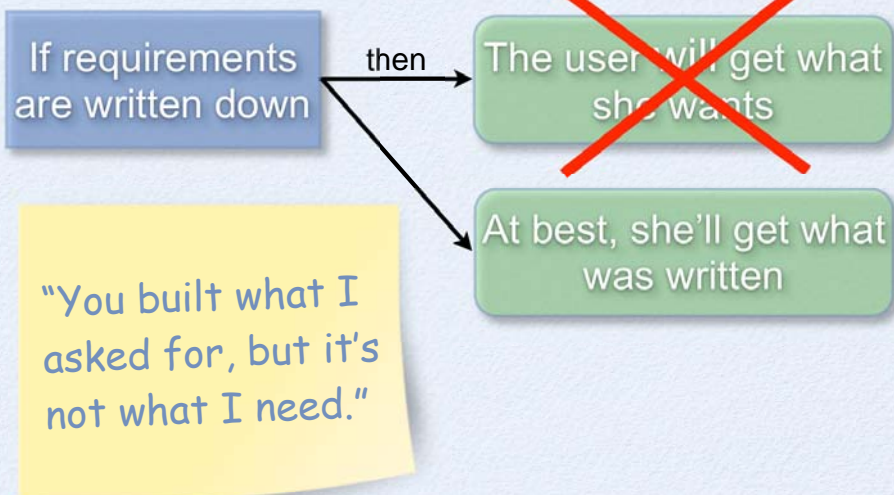
34

Words are imprecise

Entrée comes with soup or salad and bread.

↓

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

35

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?

36

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.*

Yet more reasons

- Stories support opportunistic development

  - We design solutions by moving opportunistically between top-down and bottom-up approaches[†]

- Stories support participatory design

- Stories put the focus on the user's goals

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

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).

Most importantly…

Don't forget the purpose

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

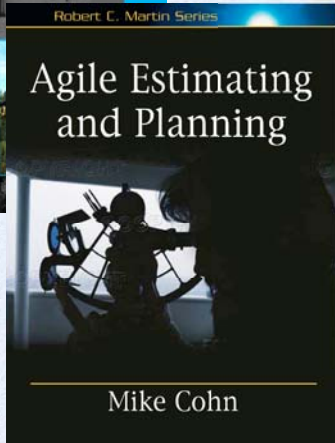# Mike Cohn contact info

mike@mountaingoatsoftware.com

www.mountaingoatsoftware.com

(720) 890-6110 (office)

(303) 810-2190 (mobile)

41