User Stories Applied

For Agile Software Development

XP Denver

April 28, 2003

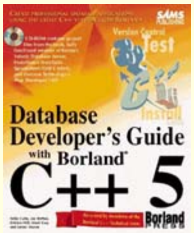
By Mike Cohn

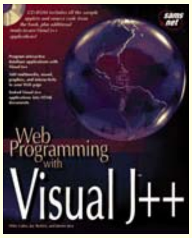
Presenter background

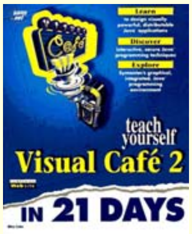
- Spent much of the last 15 years consulting and running contract development projects:
  - Viacom, Procter & Gamble, NBC, United Nations, Citibank, other smaller companies
- Have periodically taken full-time positions:
  - Genomica, McKesson, Arthur Andersen
- Diverse background across:
  - Internal software vs. Shrinkwrap products
  - Web vs. Client-server
  - Java vs. Microsoft languages
- Master's degrees in CS and Economics

Background, cont.

- Been managing projects since 1987 but remain a programmer at heart
- Books on Java, C++, and database programming.
- Articles in *IEEE Computer, STQE, C++ User's Journal*, etc.
- *User Stories Applied* will be out in early 2004







Today's agenda

- Requirements approaches
- What stories are not
- User proxies
- User roles and personas
- Gathering stories

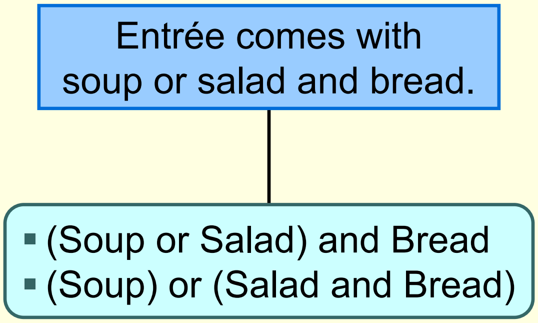
Requirements approaches

- IEEE 830
  - “The system shall…”
- Use Cases
- Scenarios
- Features from FDD
  - “Calculate the total of a sale.”
  - <action> the <result> <by|for|of|to> a(n) object
- User stories

Other approaches emphasize writing

- If we write the requirements, the customer will get exactly what she wants
  - WRONG
    - At **best**, she’ll get exactly what was written down
  - “You built what I asked for but it’s not what I want.”
- Problems with written language
  - Words are imprecise
  - Words have multiple meanings

Words are imprecise



Entrée comes with soup or salad and bread.

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

Words have multiple meanings

Buffalo buffalo buffalo.

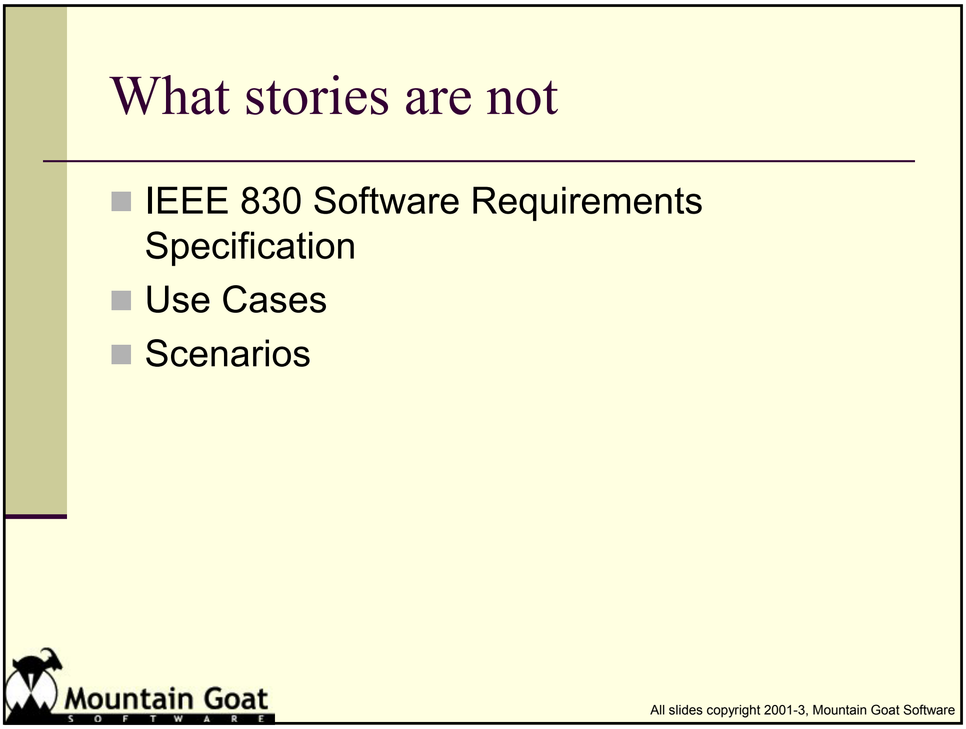
- Bison intimidate bison.

Buffalo buffalo Buffalo buffalo.

- Bison intimidate bison from Buffalo.

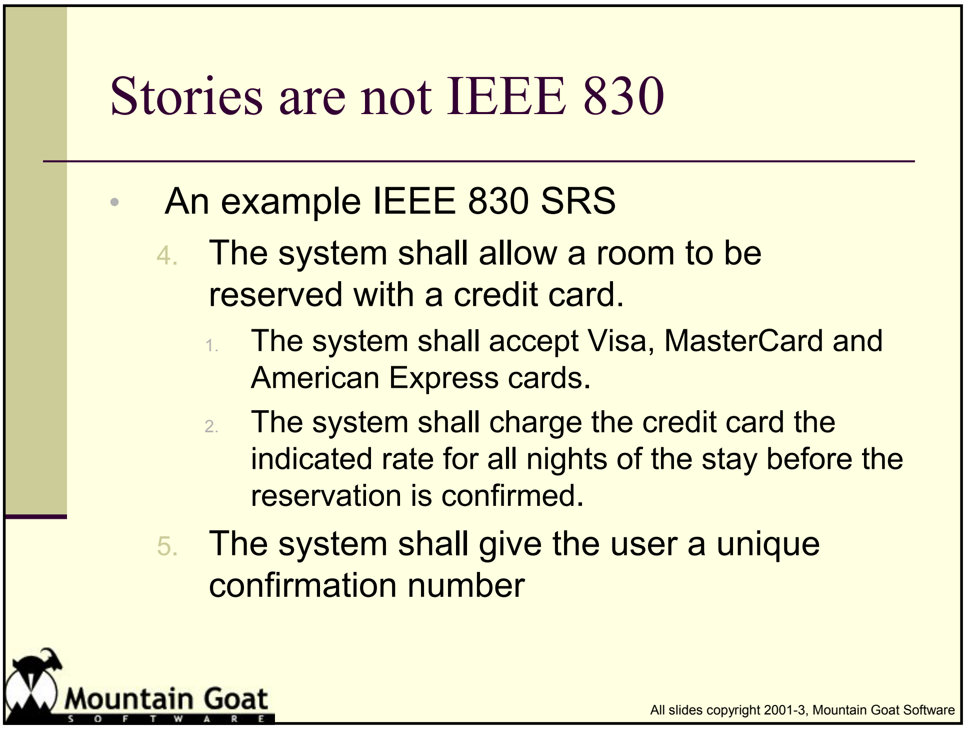
Buffalo buffalo buffalo buffalo.

- Bison intimidated by bison intimidate bison.
- Bison from Buffalo intimidate bison.



What stories are not

- IEEE 830 Software Requirements Specification
- Use Cases
- Scenarios



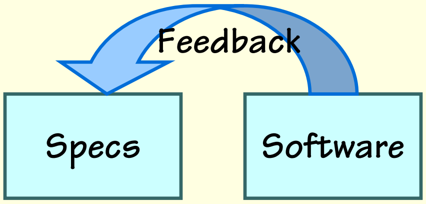
Stories are not IEEE 830

- An example IEEE 830 SRS

  4. The system shall allow a room to be reserved with a credit card.
     1. The system shall accept Visa, MasterCard and American Express cards.
     2. The system shall charge the credit card the indicated rate for all nights of the stay before the reservation is confirmed.
  5. The system shall give the user a unique confirmation number

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



- Are these changes really a "change of scope"?

All requirements are not equal

- A fluid design undermines the stability we feel
  - Humans want to feel stable
- We try to make the world stable again asap
- "Designers fix a top-level concept based on their initial understanding of a problem."
  - If they're right → "Inspiration"
  - If wrong → Painted into a corner
- "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 Specs

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).

Mountain Goat SOFTWARE

What if we had stories instead?

The product makes it easy and fast for the user to mow her lawn.

The user is comfortable while using the product.

Mountain Goat SOFTWARE

The product

Differences between use cases and stories

- Scope
  - Use case is almost always much larger
  - A story is similar to one scenario of (or path through) a use case
- Level of Completeness
  - "User stories plus acceptance tests are basically the same thing as a use case."
    - James Grenning

Differences between use cases and stories

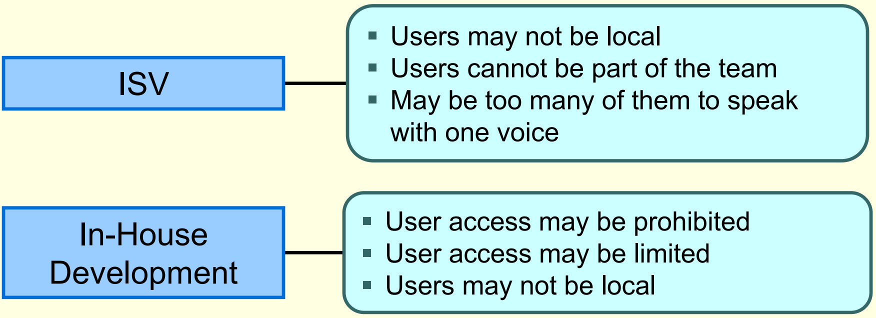
- Longevity
  - Use cases are permanent artifacts; story cards are torn up
- Purpose
  - Use cases
    - Document agreement between customer and developers
  - Stories
    - Written to facilitate release and iteration planning
    - Placeholders for future conversations

Why user stories?

- Shift focus from written to verbal communication
  - Avoid the need to put everything in writing
  - “***Represent*** customer requirements rather than ***document*** them.”*
    - Rachel Davies, “The Power of Stories,” XP 2001.
- Are relatively small pieces of functionality
  - Better for planning
- Have value to the users
  - Can be understood and prioritized by users

Who’s the User?

- To write user stories, we need a user
- Users are not as plentiful as we’d think
- Users can be hard to find:



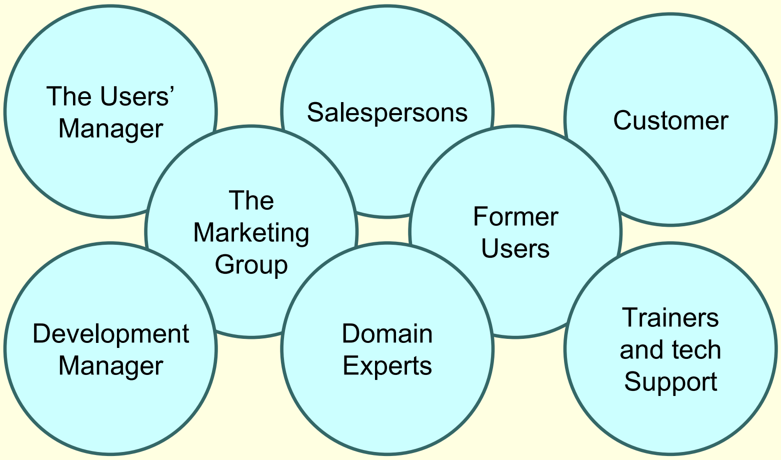
ISV

- Users may not be local
- Users cannot be part of the team
- May be too many of them to speak with one voice

In-House Development

- User access may be prohibited
- User access may be limited
- Users may not be local

User proxies



The Users’ Manager

Salespersons

Customer

The Marketing Group

Former Users

Development Manager

Domain Experts

Trainers and tech Support

What to do when access to users is restricted

- User Task Force
  - From 3 – 12 users
  - Tell the proxy that the task force is just for bouncing ideas off
  - Proxy is final decision-maker
    - Will rarely go against the opinion of the task force
  - Demo the software to this group as often as possible
    - Incorporate feedback into next iteration

What to do when there are no users

- Use more than one proxy
  - Use different types (e.g., Marketing + Domain Expert)
- Look at competing products
  - Product reviews, online newsgroups, user's guide
- Release early to real users

Can you do it yourself?

- Working with a user proxy has disadvantages

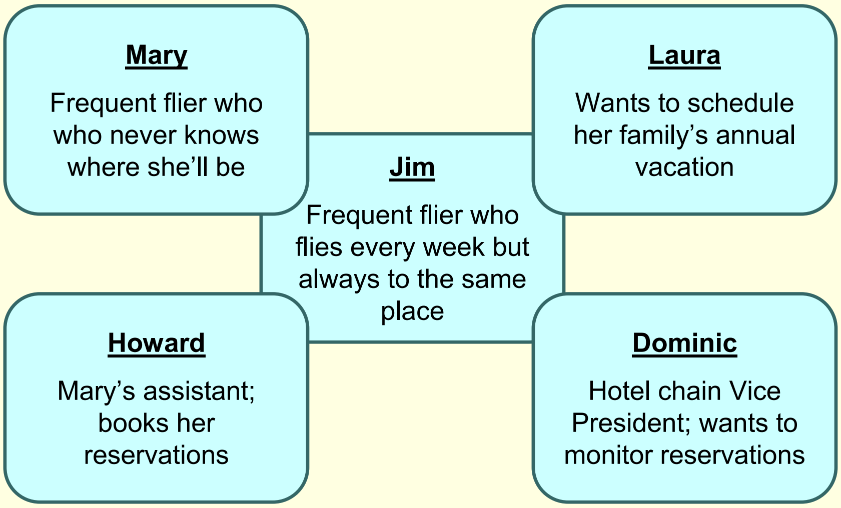
But

- A user proxy is still better than a development team taking guesses

"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?



Mary

Frequent flier who who never knows where she'll be

Laura

Wants to schedule her family's annual vacation

Jim

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

Howard

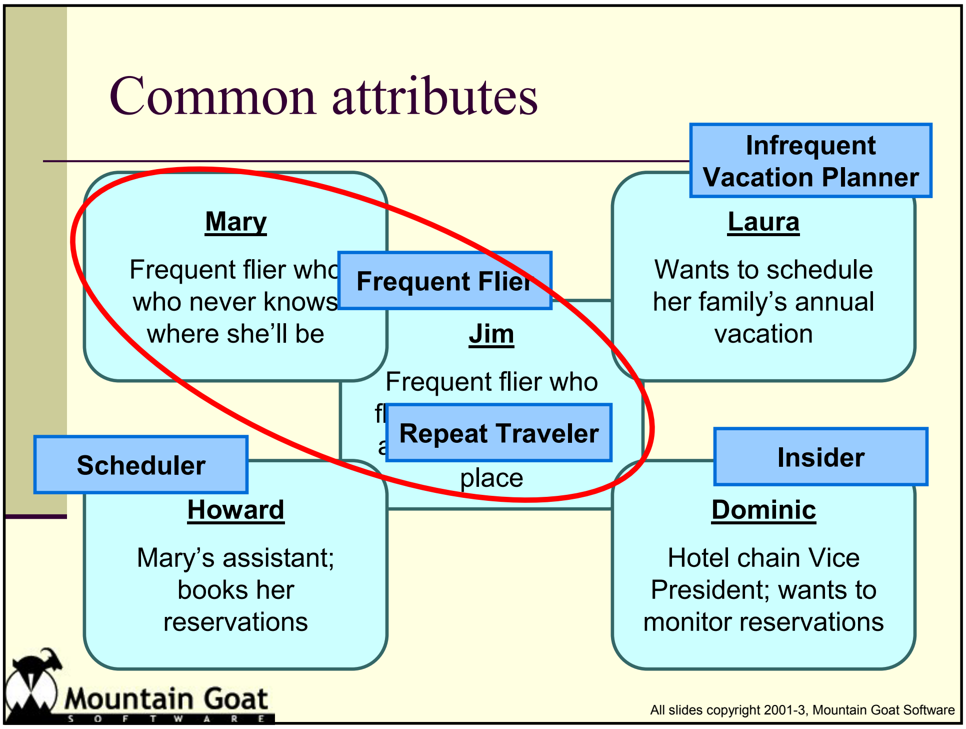
Mary's assistant; books her reservations

Dominic

Hotel chain Vice President; wants to monitor reservations

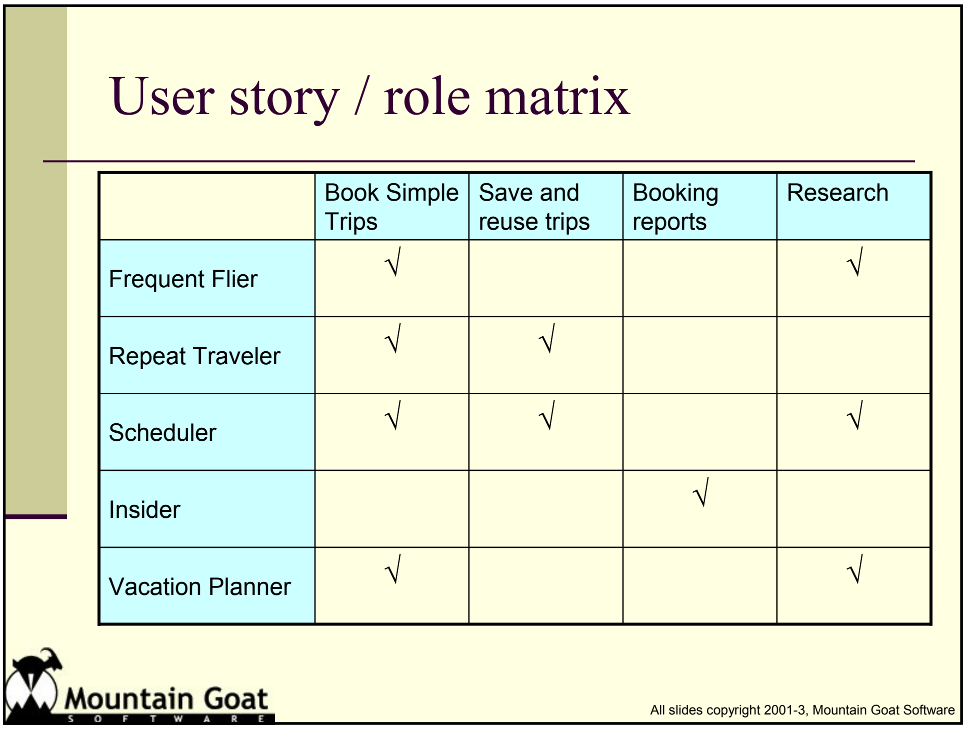
User roles

- 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.



Common attributes

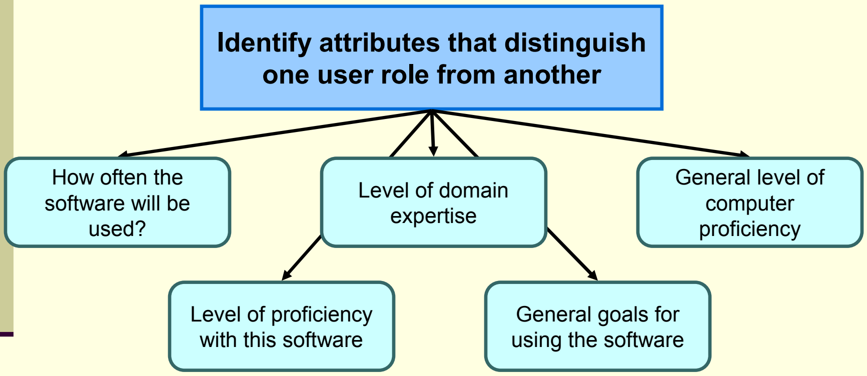
Infrequent Vacation Planner
Laura
Wants to schedule her family's annual vacation
Mary
Frequent flier who who never knows where she'll be
Frequent Flier
Jim
Frequent flier who fl a place
Repeat Traveler
Scheduler
Howard
Mary's assistant; books her reservations
Insider
Dominic
Hotel chain Vice President; wants to monitor reservations



User story / role matrix

	Book Simple Trips	Save and reuse trips	Booking reports	Research
Frequent Flier	√			√
Repeat Traveler	√	√		
Scheduler	√	√		√
Insider			√	
Vacation Planner	√			√

User role modeling


Identify attributes that distinguish one user role from another
How often the software will be used?
Level of domain expertise
General level of computer proficiency
Level of proficiency with this software
General goals for using the software

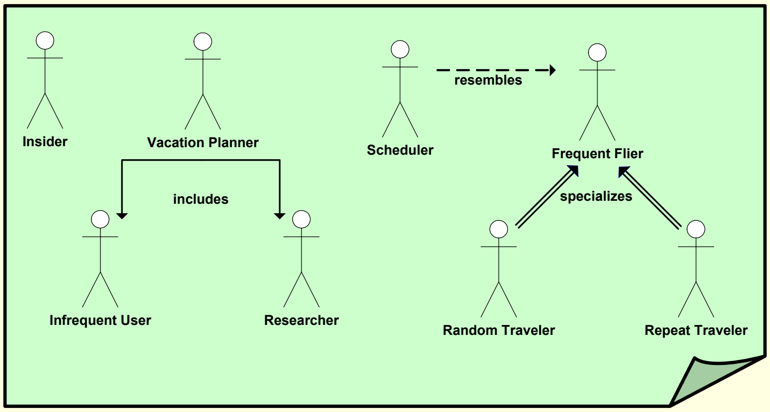
Document the user role

User Role: Infrequent Vacation Planner

Not particularly computer-savvy but quite adept at using the web. Will use the software infrequently but intensely (perhaps 5 hours to research and plan a trip). Values richness of experience (lots of content) over speed. But, software must be easy to learn and also easily recalled months later.

User Role Map

- Hard to “see” the user role inter-relationships using only text



Insider

Vacation Planner

includes

Infrequent User

Researcher

Scheduler

resembles

Frequent Flier

specializes

Random Traveler

Repeat Traveler

Personas

- A central element of Alan Cooper’s interaction design
- A persona is an imaginary representation of a user role
- A natural extension to user roles
- Generally, avoid picking personas who are real users

Add details to each persona

- Likes, dislikes
- When, where, why
- Model and make of car
- Job
  - Not “is a florist” but “works as a florist at Lake Park Florist”)
- Goals
  - Not “planning a vacation but “planning the family vacation to Yellowstone”

A sample persona

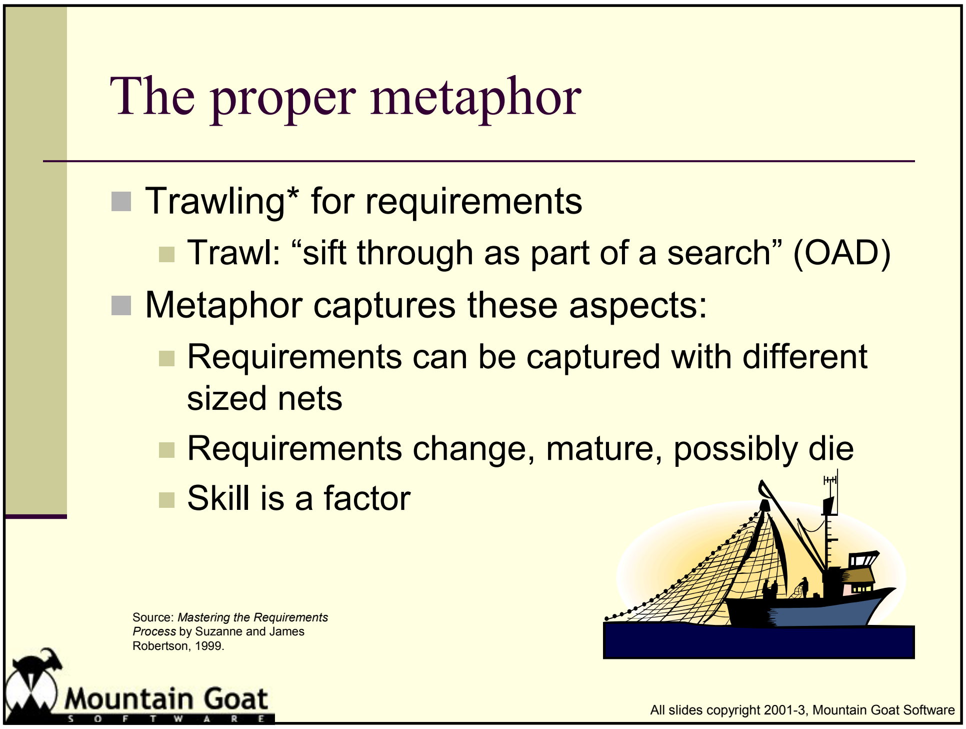
Jim lives in four bedroom house in a nice suburb north of Chicago. However, he works as a vice president of marketing in Sacramento, California. Three weeks out of every four he flies from Chicago to Sacramento on Monday morning and then flies home on Friday. The company lets him work every fourth week out of his home. Jim schedules his own flights, usually a month or more in advance. He’s partial to United Airlines but is always on the lookout for bargain fares so that the company will allow him to continue to live in Chicago. Jim quickly learns most software but becomes very impatient when he finds a bug or when a website is slow.

Using roles and personas

- Start thinking of the software as solving the needs of real people
- Avoid saying “the user” and instead say
  - “A Frequent Flier…”
  - “A Repeat Traveler…”
  - “Jim…”

Gathering stories

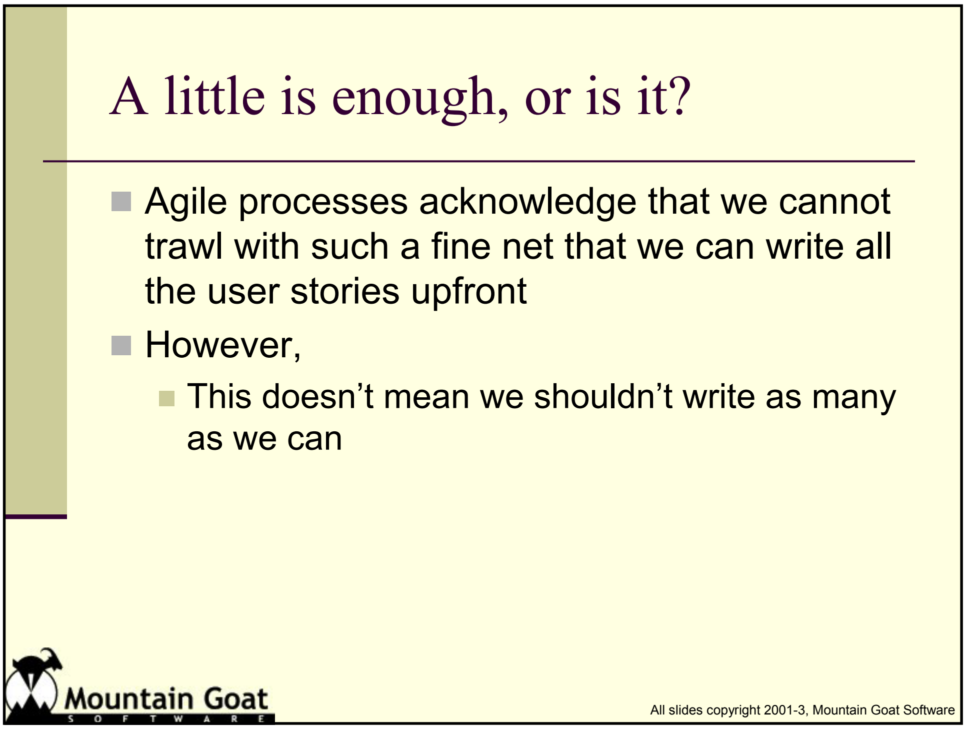
- Common metaphors for requirements are wrong
  - “Eliciting requirements”
  - “Capturing requirements”
- These metaphors imply
  - Users know the requirements but don’t want to tell us
  - Requirements need to be locked up once “captured”



The proper metaphor

- Trawling* for requirements
  - Trawl: “sift through as part of a search” (OAD)
- Metaphor captures these aspects:
  - Requirements can be captured with different sized nets
  - Requirements change, mature, possibly die
  - Skill is a factor

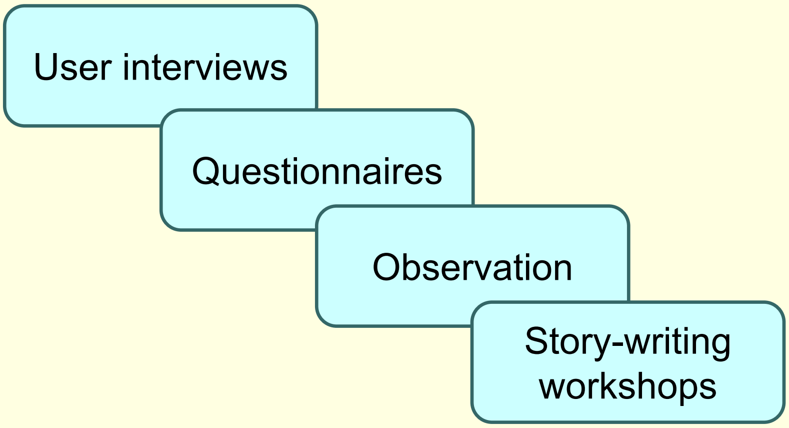
Source: *Mastering the Requirements Process* by Suzanne and James Robertson, 1999.



A little is enough, or is it?

- Agile processes acknowledge that we cannot trawl with such a fine net that we can write all the user stories upfront
- However,
  - This doesn’t mean we shouldn’t write as many as we can

Techniques for trawling for user stories


User interviews

Questionnaires

Observation

Story-writing workshops

Interviews

- Default approach taken by many teams
- Selection of interviewees is critical
  - Try to interview as many user roles as possible
- Cannot just ask “So whaddaya want?”
  - Most users are not adept at understanding their true needs
  - Having a problem does not uniquely qualify you for knowing how to solve it

Open-ended and context-free questions

- “Would you like it in a browser?”
- Two problems:
  - A closed-ended question
  - Has no context
- Instead ask:
  - “Would you like it in a browser rather than as a native Windows application even if it means reduced performance, a poorer overall user experience, and less interactivity?”
- Still, that question can be improved
  - “What would you be willing to give up in order to have it in a browser?”

Questionnaires

- Good technique for learning more about stories you already have
- If you have a large user base, great way to get information to help prioritize stories
- Not effective as a primary means of trawling for new stories

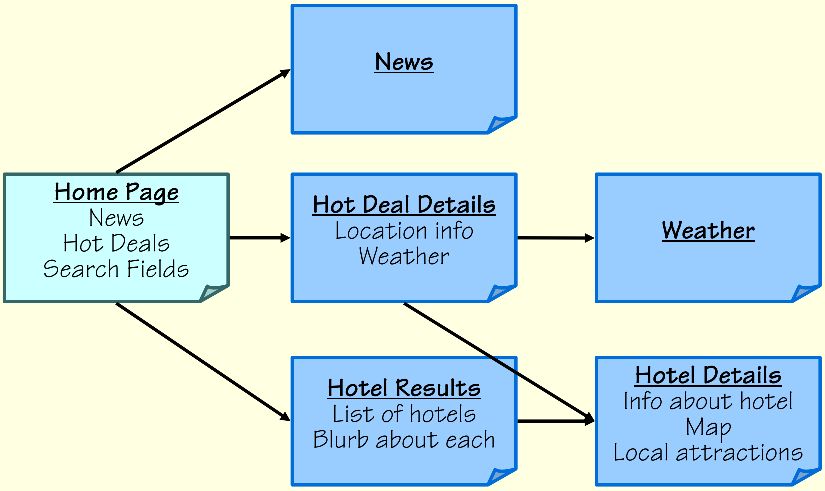
Observation

- Great way to pick up insights
- Two approaches
  - Just observe, with or without user’s knowledge
  - Have the user demonstrate to a group how she uses the software
- Example
  - Stated need:
    - “We need a large text field to summarize.”
  - Observed need:
    - Have the system record the user’s choices

Story-writing workshops

- Includes developers, users, customer, others
- Goal is to write as many stories as possible
  - Focus on quantity, not quality
  - No prioritization at this point
- Uses low-fidelity prototyping and brainstorming techniques

A low-fidelity prototype



Home Page
News
Hot Deals
Search Fields

News

Hot Deal Details
Location info
Weather

Weather

Hotel Results
List of hotels
Blurb about each

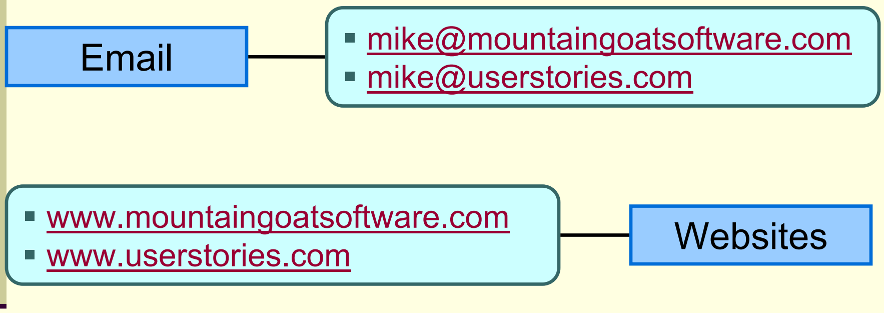
Hotel Details
Info about hotel
Map
Local attractions

Further Sources

Where to go next?

User Stories	www.userstories.com groups.yahoo.com/group/userstories
Agile in General	www.agilealliance.com
Usage-Centered Design	www.foruse.com

My contact information



Email

- mike@mountaingoatsoftware.com
- mike@userstories.com

- www.mountaingoatsoftware.com
- www.userstories.com

Websites