Oregon State University

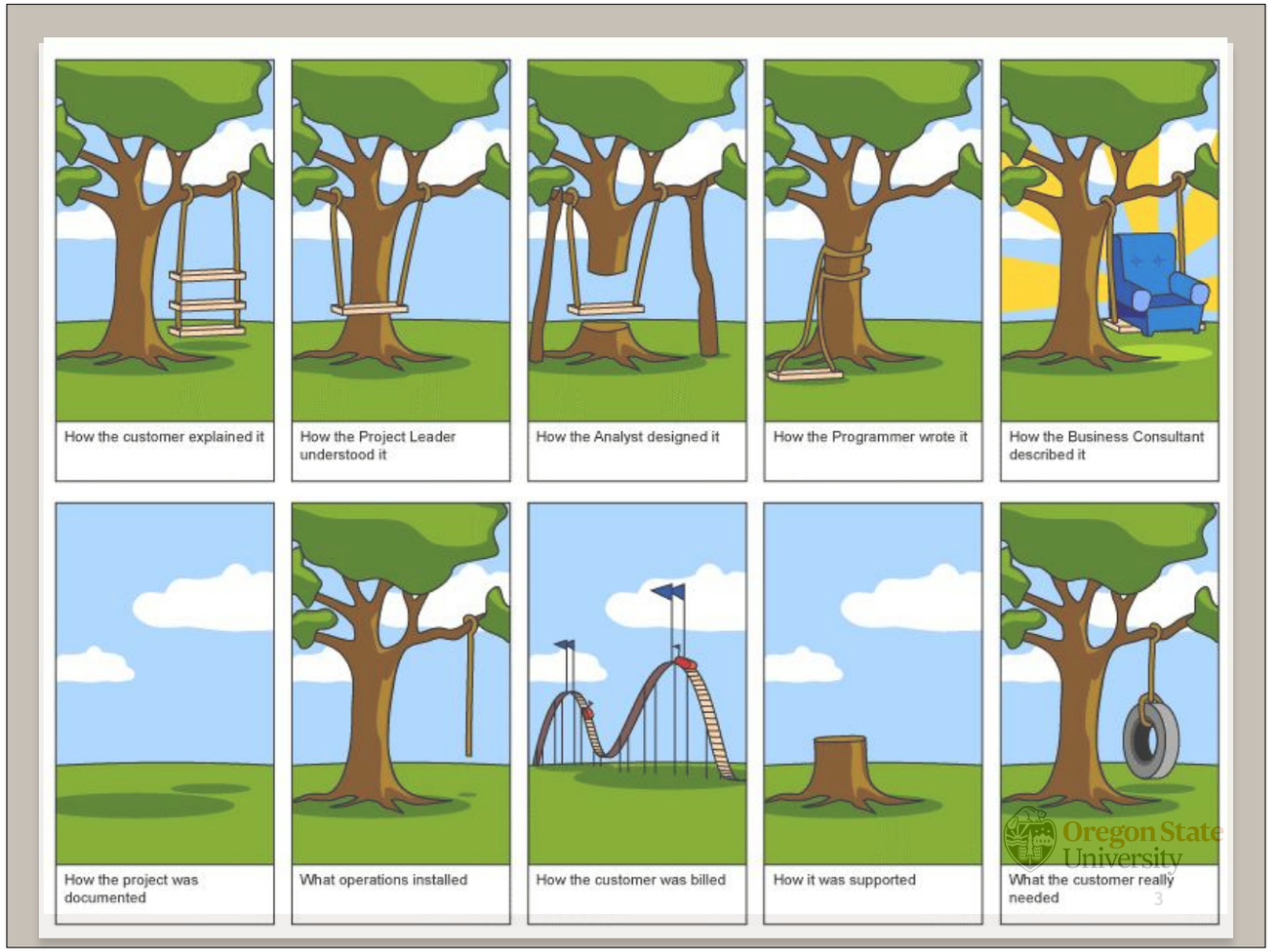
Requirements

Announcement

Next week :

- I will be traveling to conference - VL/HCC - 2017
- OO-class/lab on Tuesday
- OO-lab/project work time Thursday
- Participation quizzes
- Sprint 1 - due Friday Oct 13

Oregon State University



Requirements analysis / elicitation

Ways to figure out what the system should do:

- Get the customers to write down what they want
- Talk with customers and make some diagrams
- Watch users in “daily life” to see what they need
- Look up the requirements from a standards body
- Gather with customer & users to discuss, argue, and negotiate

Any combination, variation, or extension of the above

Oregon State University

Good requirements are…

1. Correct: They have to say the right things.
2. Consistent : They can’t contradict each other.
3. Unambiguous: Each must have 1 interpretation.
4. Complete: They cover all the important stuff.
5. Relevant: Each must meet a customer need.
6. Testable: There must be a way to tell if they are satisfied.
7. Traceable: There must be a way to determine their origin.

Types of requirements

- *Functional*
- *Non Functional (NFR)*

Oregon State University

Before we do this. What are the requirements for a video hosting site.

Non-functional requirements usually relate to quality attributes

The quality attributes of great software:

- Reliability
- Robust
- Efficiency
- Usability
- Maintainability
- Testability
- Flexibility
- Portability
- Interoperability
- Reusability

Oregon State University

- Reliability
- Efficiency
- Integrity
- Usability
- Maintainability

Examples: What quality attribute?

"The system must ask for tweet clarification within 2 minutes."

so the user is probably still online and can easily respond

"The drawbridge must rise within 1 minute."

so traffic has stops only ~ 5 minutes (1+1+ 3 for ship)

"At least 95% of the code must be Java."

because porting such applications to Linux has proven to cost only $XXXX in the past

Oregon State University

NFRs

1: usability

2. Efficiency

3. Portability

Non-functional requirements usually relate to quality attributes

The quality attributes of great software:

- **Reliability**
- Robust
- **Efficiency**
- **Usability**
- Maintainability
- **Testability**
- Flexibility
- Portability
- Interoperability
- Reusability

Oregon State University

- Reliability
- Efficiency
- Integrity
- Usability
- Maintainability

User Stories

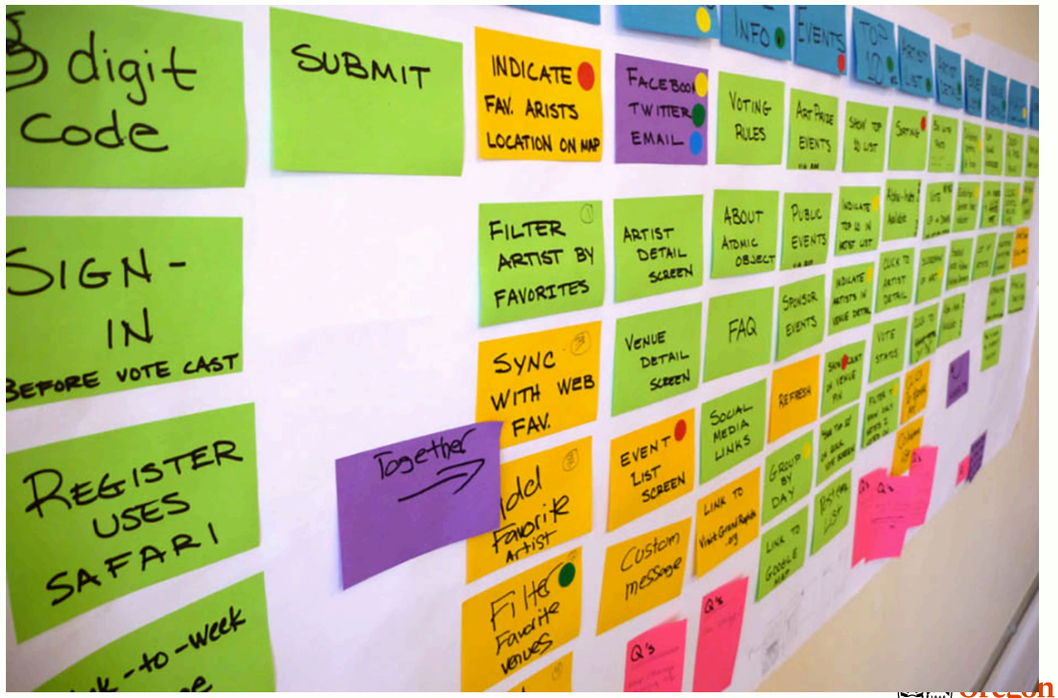
Oregon State University

How to write good user stories

- *Users come first*
  - *Use personas*
- *Create stories collaboratively*
- *Keep your stories simple and consistent (3Cs)*
- *Start with epics*
  - *Refine the stories until they are ready*
- *Use paper cards*
  - *Keep them visible & accessible*

Oregon State University

Product BackLog



https://www.scrumalliance.org/community/articles/2007/march/glossary-of-scrum-terms

User Stories (3Cs)

"…describe a small piece of system functionality, in a simple and easy to read sentence"

- The card
- The conversation
- The confirmation

http://www.subcide.com/articles/how-to-write-meaningful-user-stories/

Oregon State University

The Card

"As a [role], I want [function], so that [value]"

Often written on 3x5 card

— User story (card) for YouTube type video hosting site

Oregon State University

What is the basic card for You Tube

As a consumer, I want shopping cart functionality to easily purchase items online.
As an executive, I want to generate a report to understand which departments need to improve their productivity.

The Card

"As a [role], I want [function], so that [value]"

Often written on 3x5 card

As a Creator, I want to upload a video so that any users can view it.

As a User, I want to search by keyword to find videos that are relevant to me.

Oregon State University

What is the basic card for You Tube

As a consumer, I want shopping cart functionality to easily purchase items online.

As an executive, I want to generate a report to understand which departments need to improve their productivity.

The conversation

- An open dialog between everyone working on the project and the client
- Write conversation for: *…creator… upload a video from my local machine… any user can view it*

As a Creator, I want to upload a video from my local machine so that any users can view it.

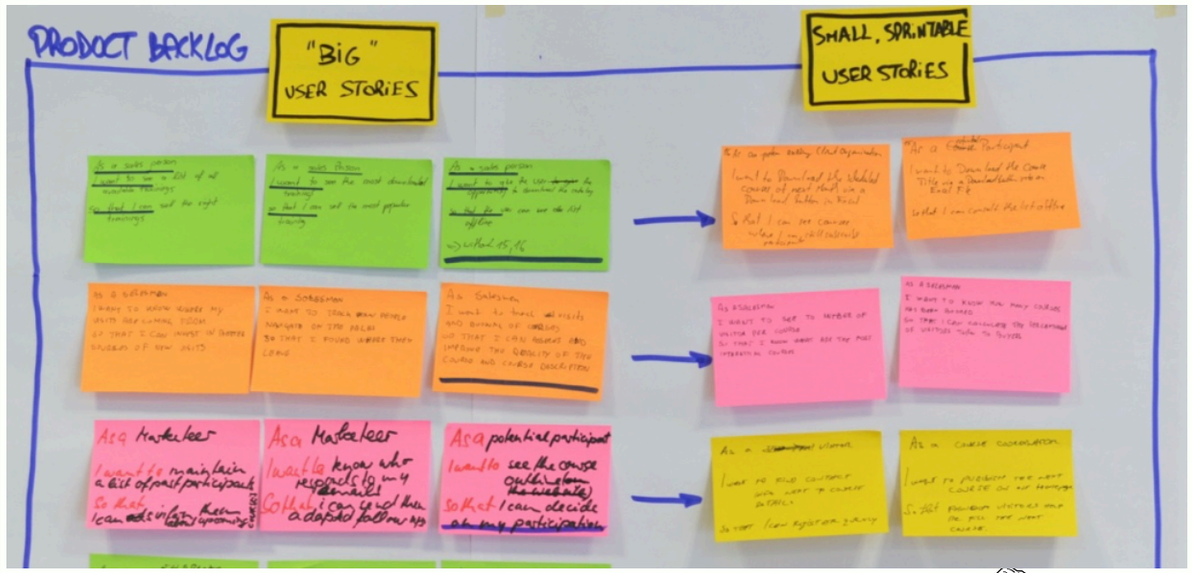
- *The "Upload" button will be a persistent item on every page of the site.*
- *Videos must not be larger than 100MB, or more than 10 minutes long.*
- *File formats can include .flv, .mov, .mp4, .avi, and .mpg.*
- *Upload progress will be shown in real time.*

As a Creator, I want to edit the video's metadata while a video is uploading, to save myself time.

- *Editable fields include Video Name, description, tags, and privacy settings.*
- *Once saved, the user will be taken to their video's dedicated page.*

conversation as an open dialogue between everyone working on the project, and the client. Anyone can raise questions, ask for things to be clarified, and the answers can be recorded down as bullet points for later reference.

Split Epic (stories) to smaller stories



The confirmation

- A test that will show when task is completed
- Could be automated, or a script
- Write conversation for: *…creator… upload a video from my local machine… any user can view it*

As a Creator, I want to upload a video from my local machine so that any users can view it.

1. *Click the "Upload" button.*
2. *Specify a video file to upload.*
    1. *Check that .flv, .mov, .mp4, .avi, and .mpg extensions are supported.*
    2. *Check that other filetypes aren't able to be uploaded.*
    3. *Check that files larger than 100MB results in an error.*
    4. *Check that movies longer than 10 mins result in an error.*
3. *Click "Upload Video".*
4. *Check that progress is displayed in real time.*

The confirmation is basically just a test case. If you're not familiar with test plans and test cases, think of a test case as a series of steps that a user must do to achieve the User Story. A test plan is a collection of test cases

Example: BattleShip game

```
User Story 1 - Easy Mode

Card: As a user I would like to have an easy mode so that I can predict the AI's moves during the course of the game.

Conversation:

Every game played in easy mode will have the AI ships located in the same positions. The AI will always fire at the user in the same pattern every game.

Confirmation:

See the "TO RUN OUR TESTS" section at the end of this page for instructions on how to run our tests. This particular user story is confirmed by the following tests in BattleshipModelTest.java:
shootAtPlayer()

Additionally, the computer ships located in BattleshipModel.java all have preset coordinates that do not change upon starting an easy mode game.

```

Oregon State University

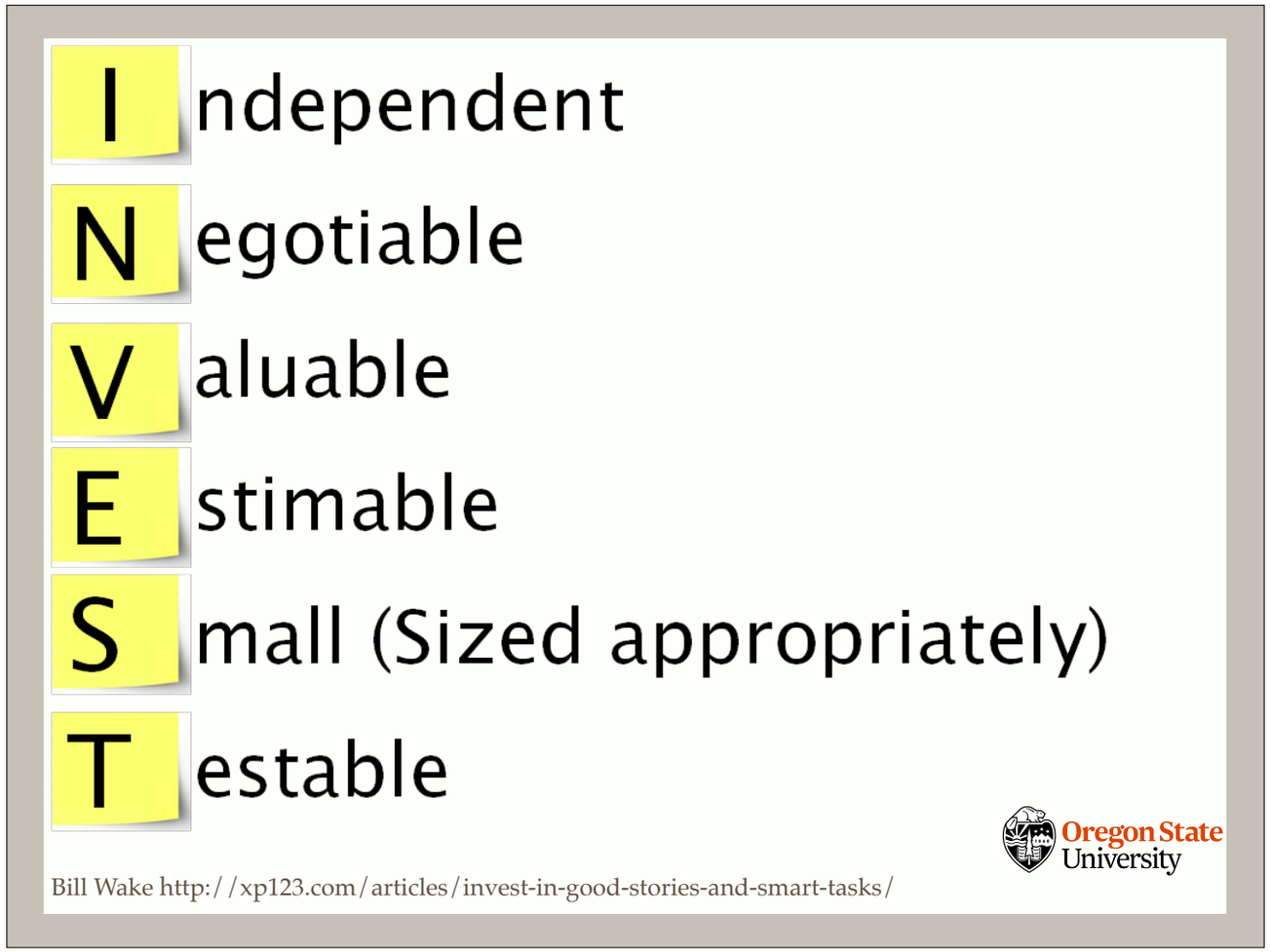
Top mistakes

Too formal or too much detail

Technical tasks masquerading as stories

Skipping the conversation

Oregon State University



Good Practices: http://www.romanpichler.com/blog/10-tips-writing-good-user-stories/

Independent

- *Schedule in any order*
- *Avoid dependency*
  - *Overlap*
  - *Order*
  - *containment*
- *Not always possible*

Oregon State University

- http://xp123.com/articles/negotiable-stories-in-the-invest-model/

- consistent and complete. Independent stories help with that too: by avoiding overlap, they reduce places where descriptions contradict each other, and they make it easier to consider whether we've described everything we need.

- Overlapping: When stories overlap, it's hard to ensure that everything is covered at least once, and we risk confusion when things are covered more than once.
- **User sends & replies to messages. Users sends & receives messages. & is suspicious**

- Order dependencies: the skinniest possible version of account management" so focus on important aspects.
  User needs an account before they can send email

- Containment – depth first, hard to schedule. Although, helps undesrstand sets of stories

Negotiable

- Details to be negotiated during development
- Good Story captures the Essence, not the details

Oregon State University

- High-level stories, written from the perspective of the actors that use the system, define capabilities of the system without over-constraining the implementation. “As a Fulfilments officer I want to send a book and receipt” can be implemented in different ways

- Negotiable stories help even in ambiguous situations; we can work with high-level descriptions early, and build details as we go.
- By starting with stories at a high level, expanding details as necessary, and leaving room to adjust as we learn more, we can more easily evolve to a solution that balances all our needs.

Valuable

- This story needs to have value to someone (hopefully the customer)
- Especially relevant to splitting up issues

Oregon State University

- Different types of values for different types of roles.
- But, don’t get disconnected from the user/purchaser.
- Who gets the benefit of the software we create? (One person can fill several of these roles, and this is not an exhaustive list.)

Estimable

"Plans are nothing, Planning is everything" -Dwight D. Eisenhower

- *Helps keep the size small (story points)*
  - *Size, cost, time to deliver*
- *Ensure we negotiated correctly*

- *How can you estimate?*
- *Why is it hard?*

Oregon State University

How Estimate: expert opinion, analogy, decomposition, formula, sample

Why hard: domain, level of innovation, details are not fully known, dependency to other stories, team, technology, approach to solution, relationship to code, rate of change, overheads

Small (Sized appropriately)

- *Fit on 3x5 card*
- *at most two person-weeks of work*
- *Too big == unable to estimate*

Oregon State University

Role-action-context: *Example*: Customer purchases item
Example: Customer purchases item with debit card

Testable

- Ensures Understanding of Task
- We know when we can mark task “Done”
- Unable to test == do not understand

Oregon State University

Your turn

Write down 1 user story for the banking app

Simplify:
https://vimeo.com/41800652

Oregon State University

Resources on Canvas

- *Invest Model* *(how to write meaningful stories)*
- *User story examples and counter-examples*

Oregon State University

Quiz - graded

- *Get your clickers*
- *Channel 62*
- *5 questions*
- *Time = 45 seconds per question*

Oregon State University