# W04-2:
# User Stories Pt. 2

Dr. Alex Adkins
CPSC 3720: Software Engineering

# Game: Drawing!

- Follow the provided instructions
- Do your best to share the various colored pencils
- Use a color that is as close to the instructions as possible
- 10 minutes

# The Instructions:

Draw a beautiful summer meadow with different colored flowers, some green grass, cows, birds, dogs, and a shining sun.

Draw a beautiful summer meadow with:

- Green grass
- 10 blue flowers with 5 petals each
- 5 pink flowers with 6 petals each
- 13 red flowers with 6 petals each
- 2 cows with 3 brown spots
- 1 cow with 5 black spots
- 2 white and brown dogs
- 2 birds in the upper left corner
- 3 birds in the middle
- One sun to the right with 5 sunbeams

checkin.cs.clemson.edu:

# Discussion

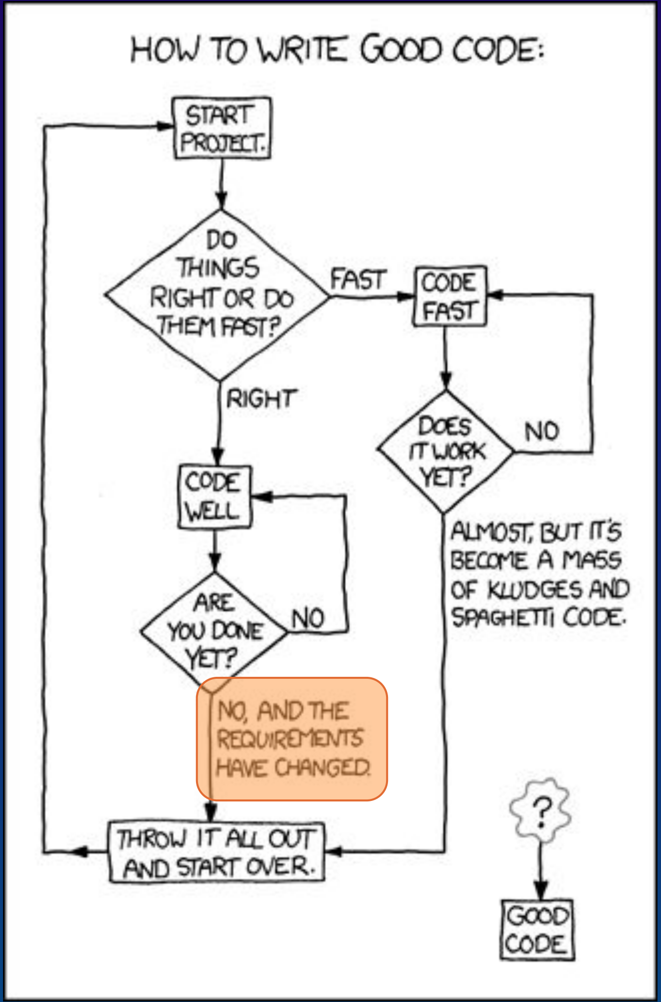
As a class…

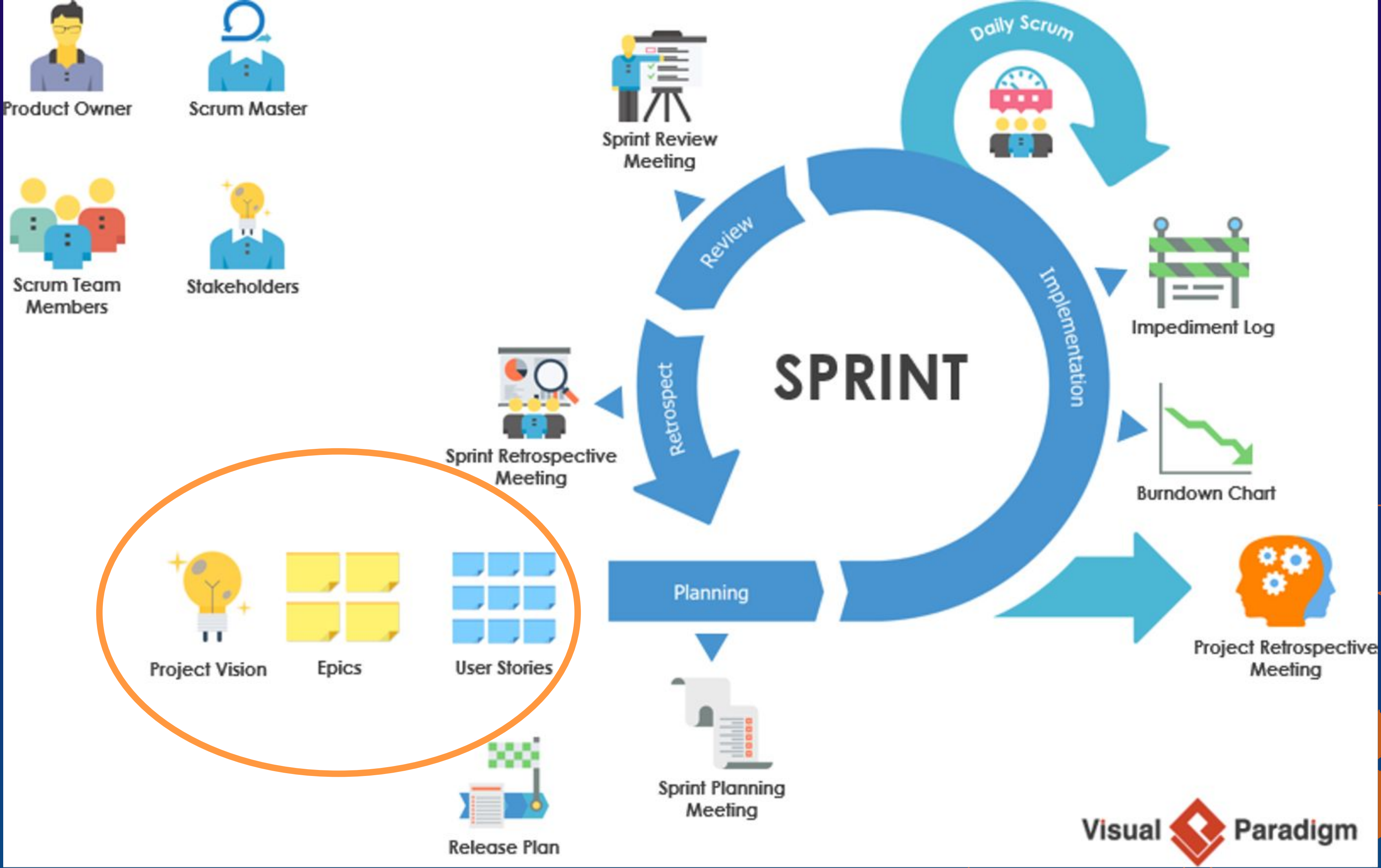
- What side of the room produced the better "meadows?"
- What design tools does your table's instructions represent? The other tables'?

# Discussion

At your tables…

- Why User Stories vs. Requirements Specs?


HOW TO WRITE GOOD CODE:
START PROJECT.
DO THINGS RIGHT OR DO THEM FAST?
FAST
CODE FAST
DOES IT WORK YET?
NO
ALMOST, BUT IT'S BECOME A MASS OF KLUDGES AND SPAGHETTI CODE.
RIGHT
CODE WELL
ARE YOU DONE YET?
NO
NO, AND THE REQUIREMENTS HAVE CHANGED.
THROW IT ALL OUT AND START OVER.
?
GOOD CODE


Product Owner
Scrum Master
Scrum Team Members
Stakeholders
Sprint Review Meeting
Daily Scrum
Review
Implementation
Impediment Log
SPRINT
Retrospect
Sprint Retrospective Meeting
Burndown Chart
Project Vision
Epics
User Stories
Planning
Project Retrospective Meeting
Sprint Planning Meeting
Release Plan
Visual Paradigm

# Today’s Objectives

- Game!
- RECAP:
    - Why Create User Stories?
    - The Three C’s
- INVEST to create great stories
- ~~Memes!~~ Logos!

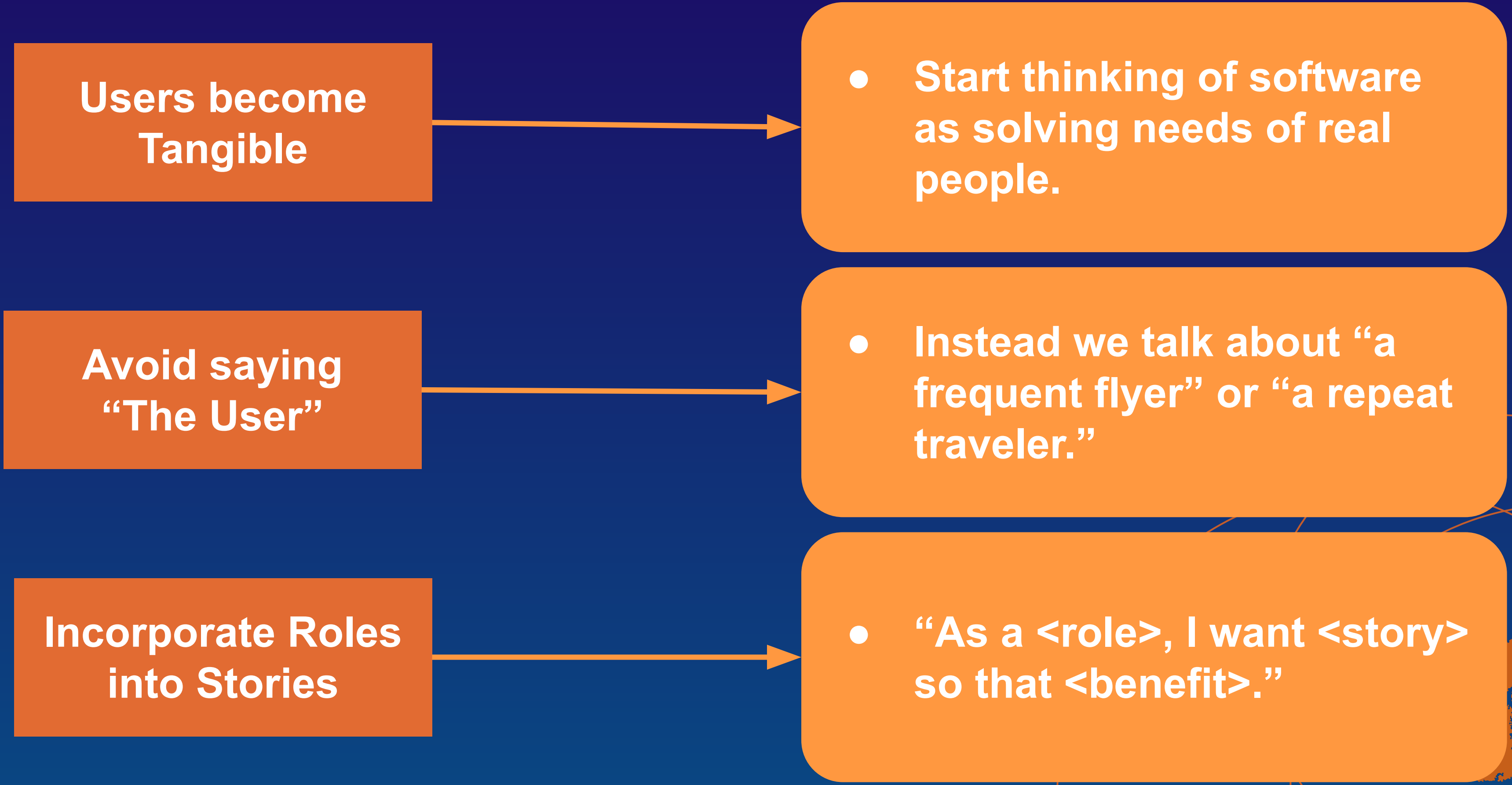
# User Stories

# User “Roles”

- Users vary by:
  - How they use the software
  - Their backgrounds
  - Their familiarity with software/computers
  - What they use the software for

- A User Role is a collection of defining attributes that characterize a population of users and their intended interactions with the system.
  - Source: *Software for Use* by Constantine and Lockwood (1999)

# Advantages of using Users/Roles



Users become Tangible

- Start thinking of software as solving needs of real people.

Avoid saying “The User”

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

Incorporate Roles into Stories

- “As a <role>, I want <story> so that <benefit>.”

# What are User Stories?

## User Stories:

- short simple descriptions of a features
- told from the perspective of the person who desires the new capability
  - usually a user or customer of the system

## User Story Template:

As a <type of user/user role>, I want <some goal> so that <some reason>

**WHO** **WHAT** **WHY**

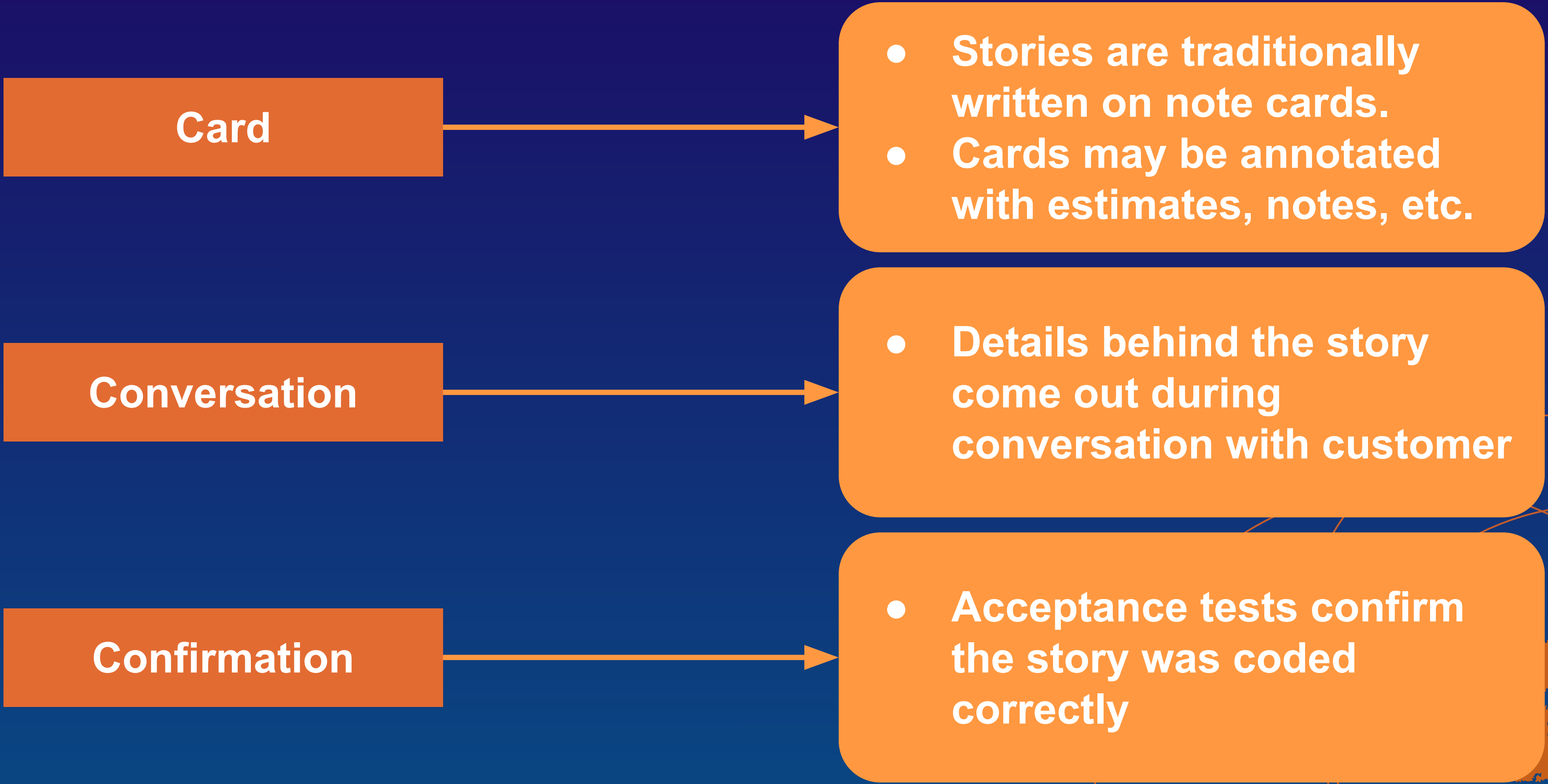
# Why User Stories?

- Stores are more comprehensible
  - Developers and customers understand them
  - People can understand them better as stories

- Stories support iterative development

- Stories support participatory design
  - The users of the system become part of the team designing the behavior of the system

# The Three C's of User Stories



Card

- Stories are traditionally written on note cards.
- Cards may be annotated with estimates, notes, etc.

Conversation

- Details behind the story come out during conversation with customer

Confirmation

- Acceptance tests confirm the story was coded correctly

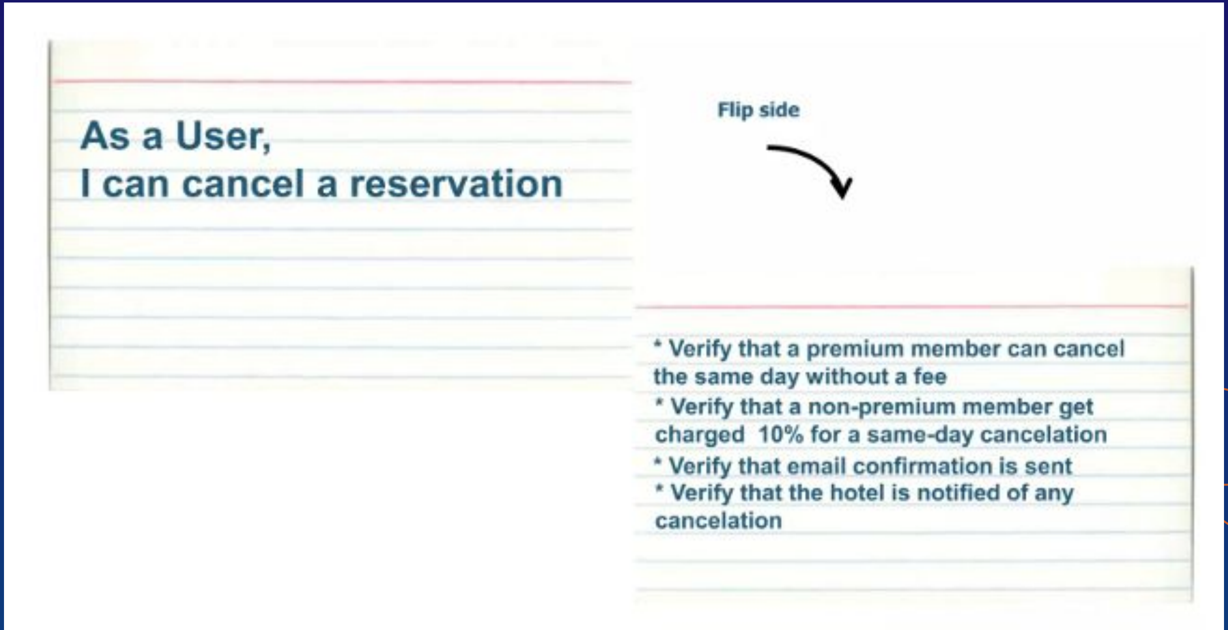
# Writing User Stories: Card & Confirmation

The story should answer:

**Who** – who the user story is for (**As a User**)

**What** – the functionality that the user story implements (**I want**)

**Why** – the reason the user needs the user story (**so that**)

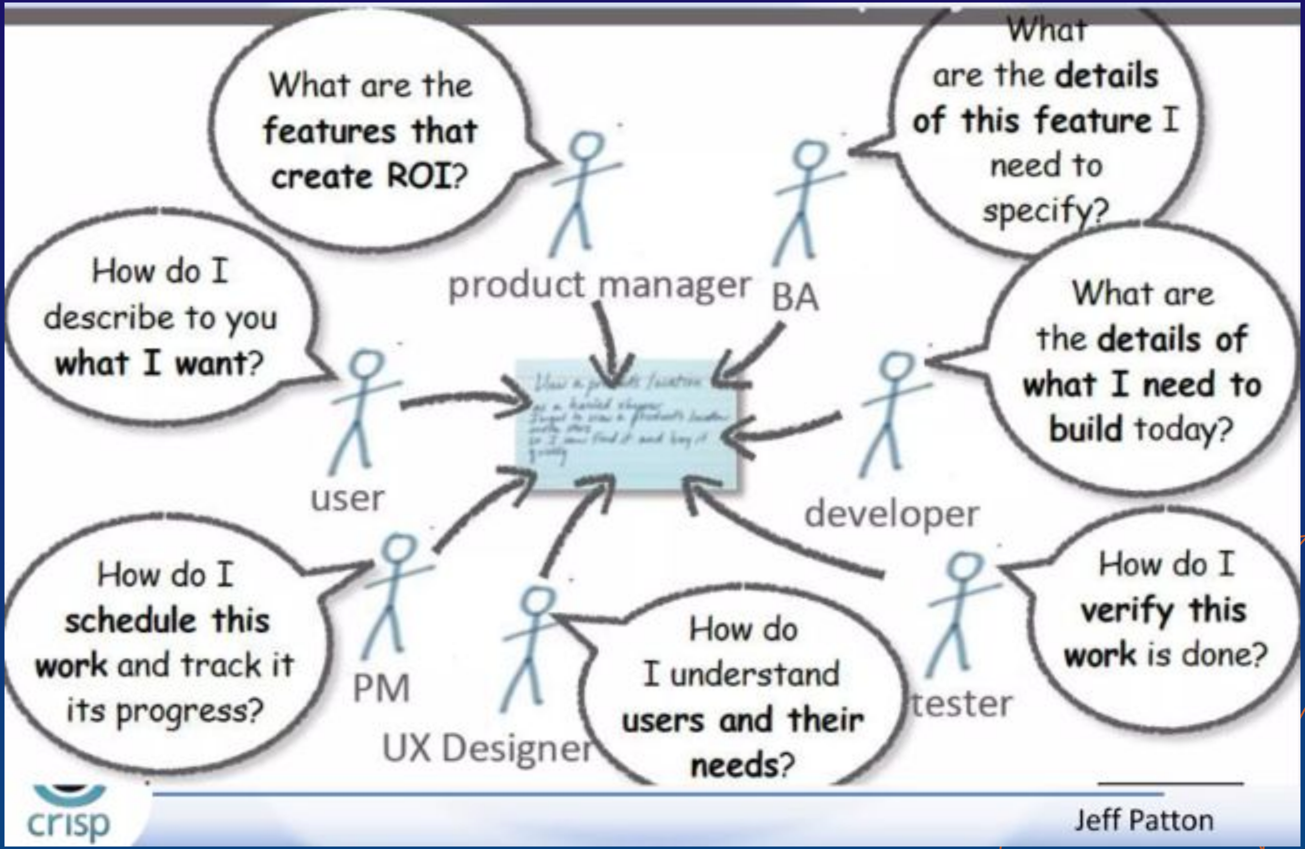


As a User,
I can cancel a reservation

Flip side

* Verify that a premium member can cancel the same day without a fee
* Verify that a non-premium member get charged 10% for a same-day cancelation
* Verify that email confirmation is sent
* Verify that the hotel is notified of any cancelation

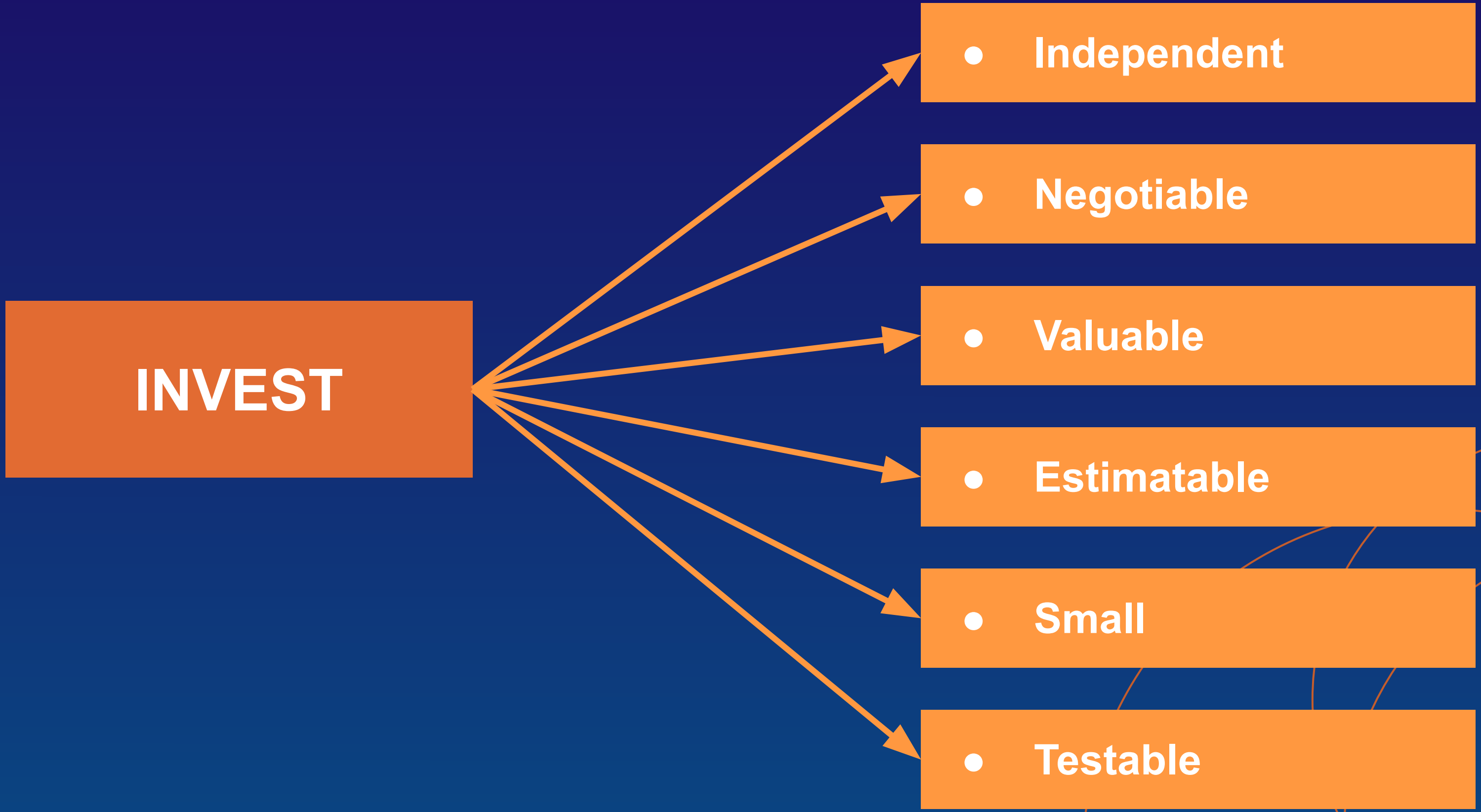
# Writing User Stories: Conversation


What are the features that create ROI?
product manager
What are the details of this feature I need to specify?
BA
How do I describe to you what I want?
user
What are the details of what I need to build today?
developer
How do I schedule this work and track it its progress?
PM
UX Designer
How do I understand users and their needs?
tester
How do I verify this work is done?
crisp

Jeff Patton

# INVEST

# INVEST to create good stories!



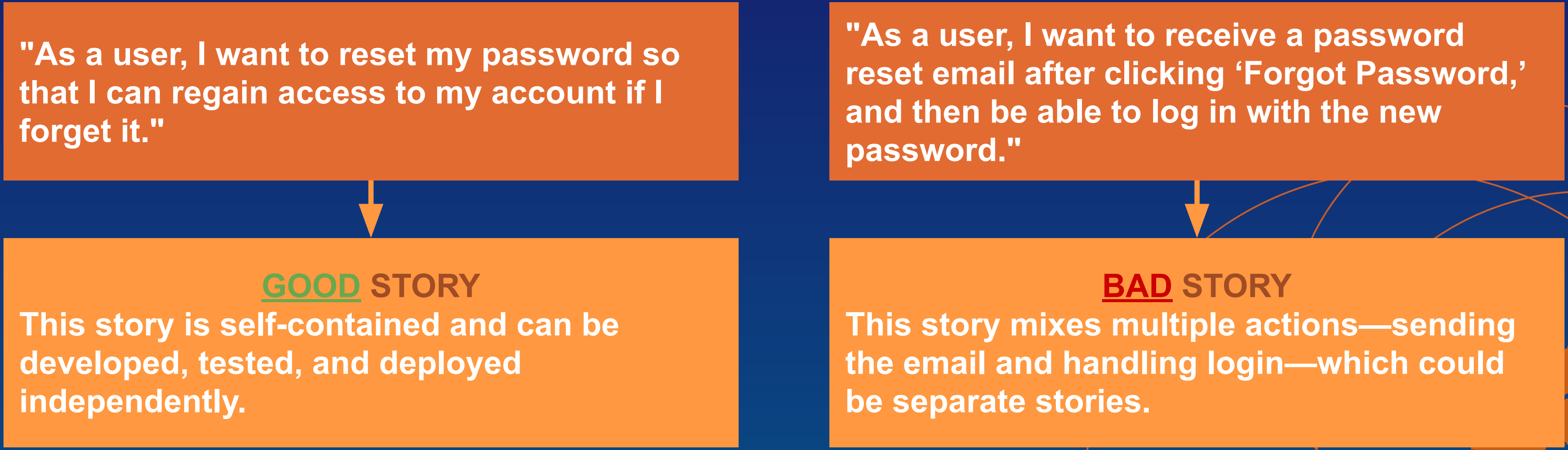
INVEST

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

# INVEST: Independent

- Avoid introducing dependencies
  - Leads to difficulty prioritizing and planning

In terms of **Independent**, are these stories good or bad? Why?



**"As a user, I want to reset my password so that I can regain access to my account if I forget it."**

**GOOD STORY**
**This story is self-contained and can be developed, tested, and deployed independently.**

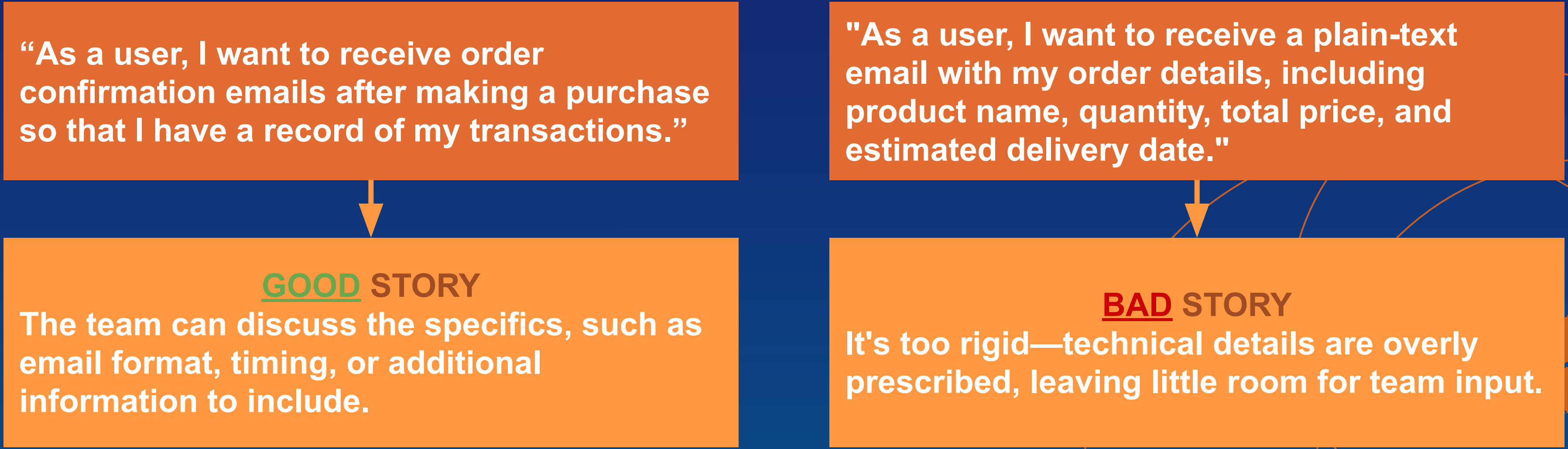
**"As a user, I want to receive a password reset email after clicking 'Forgot Password,' and then be able to log in with the new password."**

**BAD STORY**
**This story mixes multiple actions—sending the email and handling login—which could be separate stories.**

# INVEST: Negotiable

- Stories are not contracts
  - They need flexibility so that we can adjust what gets implemented
- Too much detail can give an impression of completeness or precision

In terms of **Negotiable**, are these stories good or bad? Why?



**“As a user, I want to receive order confirmation emails after making a purchase so that I have a record of my transactions.”**

**GOOD STORY**

**The team can discuss the specifics, such as email format, timing, or additional information to include.**

**"As a user, I want to receive a plain-text email with my order details, including product name, quantity, total price, and estimated delivery date."**

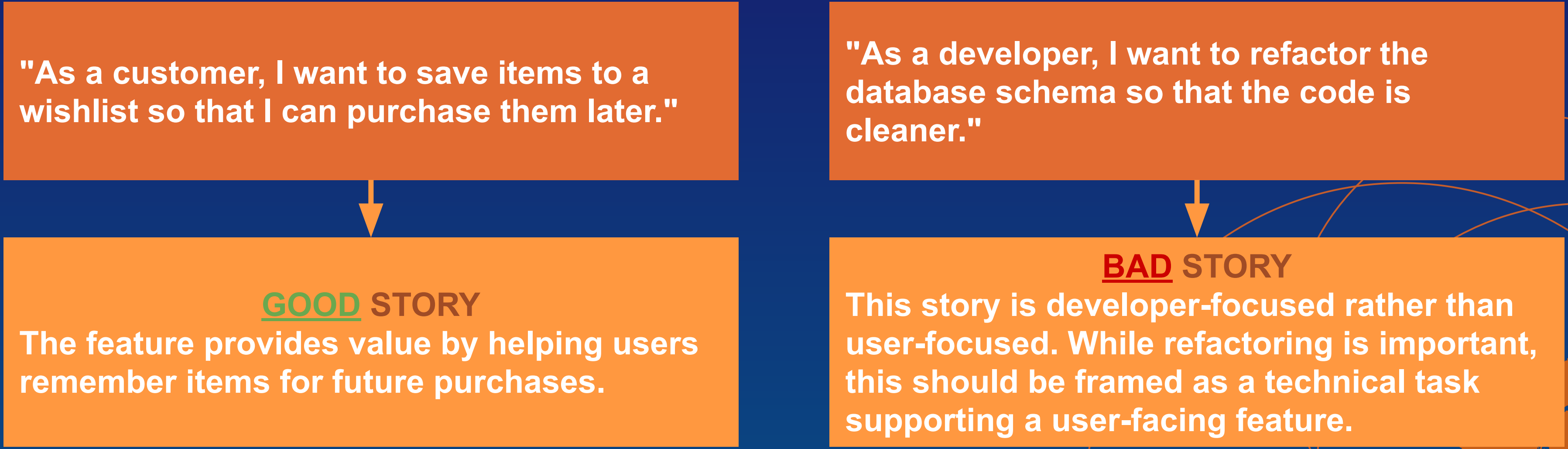
**BAD STORY**

**It's too rigid—technical details are overly prescribed, leaving little room for team input.**

# INVEST: Valuable

- Stories must provide a clear benefit to the users or the purchasers

In terms of **Valuable**, are these stories good or bad? Why?



**"As a customer, I want to save items to a wishlist so that I can purchase them later."**

**GOOD STORY**
**The feature provides value by helping users remember items for future purchases.**

**"As a developer, I want to refactor the database schema so that the code is cleaner."**

**BAD STORY**
**This story is developer-focused rather than user-focused. While refactoring is important, this should be framed as a technical task supporting a user-facing feature.**

# INVEST: Estimatable

- Stories should be small and well defined to easily estimate effort
  - Stories may not be estimatable if:
    - Developers lack domain or technical knowledge
    - The story is too big or vague

In terms of **Estimatable**, are these stories good or bad? Why?

**"As a user, I want to update my profile picture so that my account displays a personalized image."**

**GOOD STORY**
**The scope is clear, and the team can estimate the work needed to add an image upload feature.**

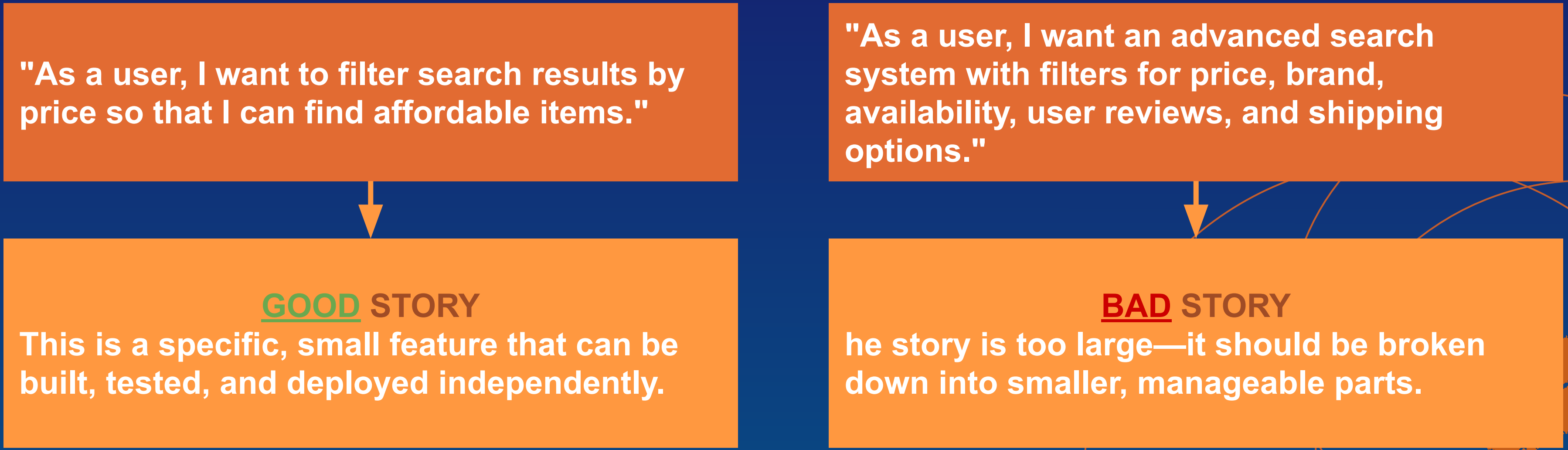
**"As a user, I want a completely redesigned profile page with a better UI."**

**BAD STORY**
**The story is too vague—"better UI" is subjective, and the scope is unclear, making it difficult to estimate.**

# INVEST: Small

- Stories should be able to be completed within a single sprint
  - Large stories are hard to estimate and hard to plan
    - They don’t fit well into single sprints

In terms of **Small**, are these stories good or bad? Why?



**"As a user, I want to filter search results by price so that I can find affordable items."**

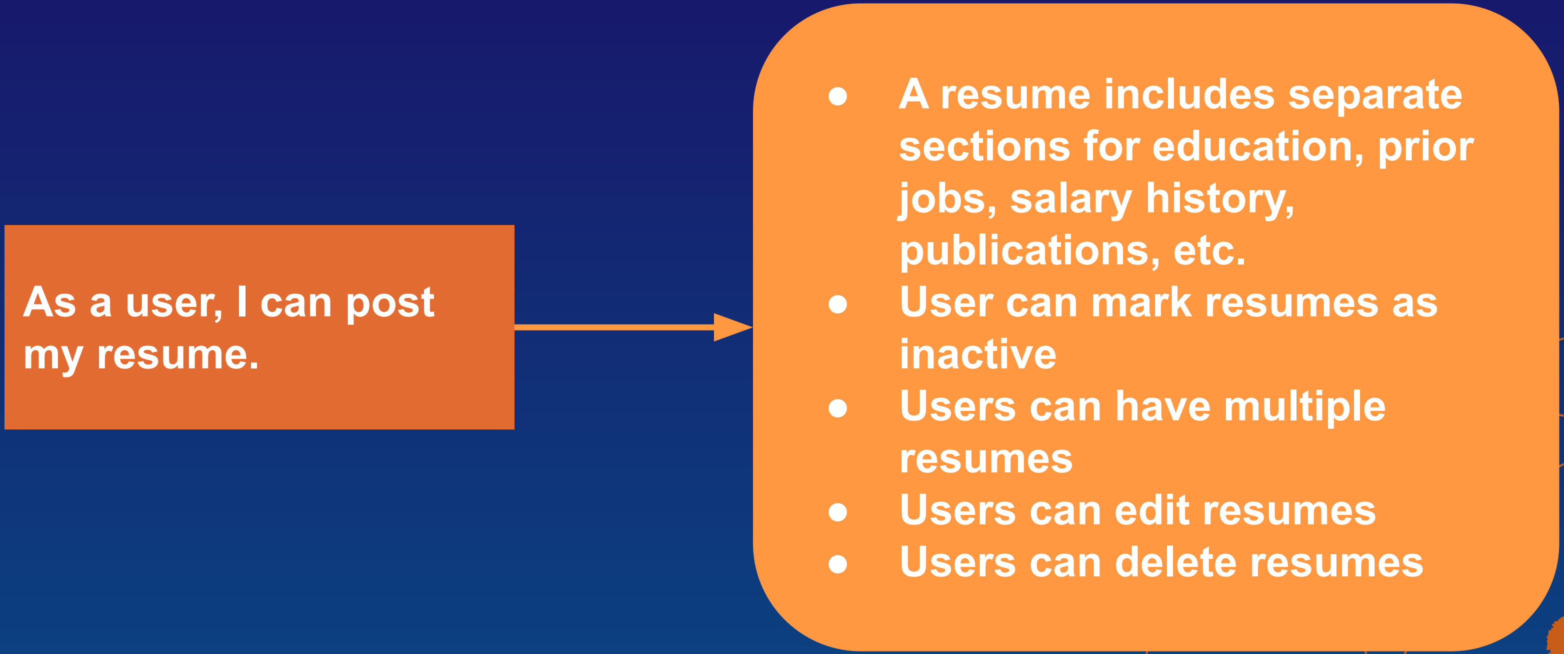
**GOOD STORY**
**This is a specific, small feature that can be built, tested, and deployed independently.**

**"As a user, I want an advanced search system with filters for price, brand, availability, user reviews, and shipping options."**

**BAD STORY**
**he story is too large—it should be broken down into smaller, manageable parts.**

# INVEST: Small - Compound Stories

- Stories can often hide a great number of assumptions:



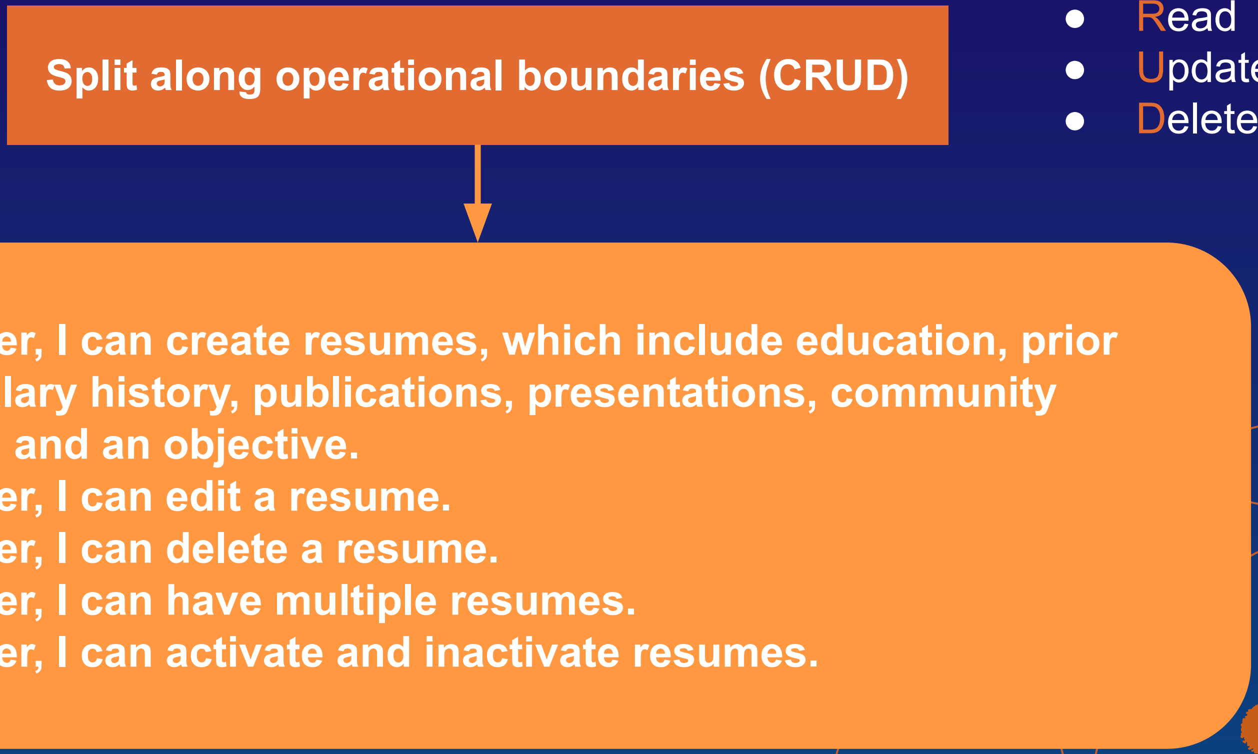
**As a user, I can post my resume.**

- **A resume includes separate sections for education, prior jobs, salary history, publications, etc.**
- **User can mark resumes as inactive**
- **Users can have multiple resumes**
- **Users can edit resumes**
- **Users can delete resumes**

# INVEST: Small - Compound Stories

**CRUD:**

- Create
- Read
- Update
- Delete



**Split along operational boundaries (CRUD)**

- **As a user, I can create resumes, which include education, prior jobs, salary history, publications, presentations, community service, and an objective.**
- **As a user, I can edit a resume.**
- **As a user, I can delete a resume.**
- **As a user, I can have multiple resumes.**
- **As a user, I can activate and inactivate resumes.**

# INVEST: Small - Compound Stories

**Split along data boundaries**

- **As a user, I can add and edit educational information on a resume.**
- **As a user, I can add and edit prior jobs on a resume.**
- **As a user, I can add and edit salary history on a resume.**
- **As a user, I can delete a resume.**
- **As a user, I can have multiple resumes.**
- **As a user, I can activate and inactivate resumes.**

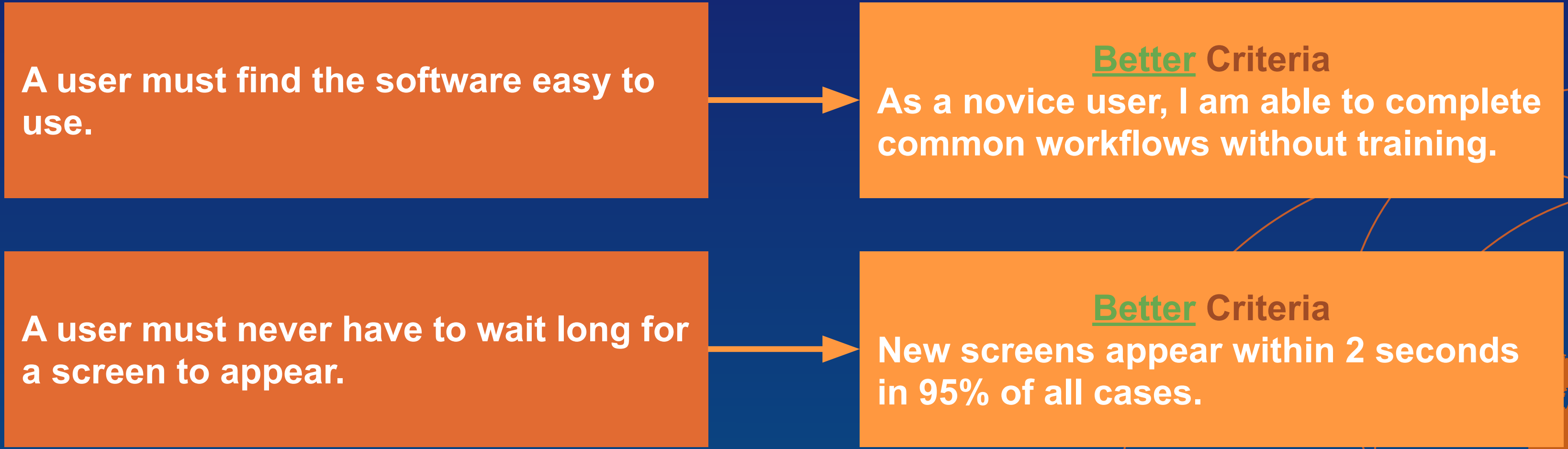
# INVEST: Testable

**SAM:**

- Specific
- Attainable
- Measurable

- Stories should have clear and testable acceptance criteria
- Tests demonstrate that a story meets the customer’s expectations
  - Strive for 90%+ testing automation

How can we improve these acceptance criteria?



**A user must find the software easy to use.**

**Better Criteria**
**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.**

**Better Criteria**
**New screens appear within 2 seconds in 95% of all cases.**

# User Stories: General Guidelines

## INVEST

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

- S.A.M. (helps with testable)
  - Specific
  - Attainable
  - Measurable

## Story Hierarchy

- Product: the largest chunk of value
- Epic: a medium chunk of value
  - larger than a story but smaller than the whole product
- User Story: smallest piece of value
  - can be implemented in at most a **week or two**
  - **broken down into tasks for each sprint**

## Stories should answer: Who, What, Why?

- Who - the the user story is for (As a User)
- What - the Functionality that the user story implements (I want)
- Why - the reason the user needs the user story (so that)

# Let’s practice!

**Automated Teller Machine (ATM)**

- Deposit money
- Withdrawal Cash
- Check account balances

# Let's Practice! (10 minutes)

You have the following Epic:

***As an ATM User, I can <u>deposit</u> cash, so that I can have increased funds in the bank.***

What are some stories you can derive from this Epic?

Each table work as a team to come up with at least 3 stories for this Epic on the index cards. Write your table number on the card. Follow this format:

**Front:** **As a ___________, I want to ____________, so that_______________**

**Back:** **Story Acceptance Criteria**

# Up Next!

- Project Skill Survey completed by this Friday at end of day
  - Find it in Canvas under Week 4
  - Worth 5 points!

- Project Team Requests by end of day Friday via email (NO EXCEPTIONS)
  - Have someone you want to work with? Email Dr. Alex by EOD Friday

- Project Kickoff next week!

# Sources

Much of the presentation content was developed using slides and information provided and permitted for this use by Mountain Goat Software: www.mountaingoatsoftware.com.

The company (and founder Mike Cohn) are great sources for Agile best practices.