# User Stories

- Similar to Use Cases but not the same
  - User stories are centered on the result and the benefit of the thing you're describing, whereas use cases are more granular, and describe how your system will act. From: http://www.boost.co.nz/blog/2012/01/use-cases-or-user-stories/

- Use cases: actors – scope – goals – steps – success
  - Details of most important requirements worked out ahead of time to ensure that everyone is on the same page
  - Useful for groups of similar stories and describing overall system
    - Use cases decompose stories into actions in the system

- User stories: scope of a feature + acceptance criteria
  - Each feature is captured as a story; stories easily prioritized
  - A story is a place holder for discussion and planning poker in a sprint

See **recommended reading** links for examples and suggestions

# Writing Good User Stories

- Its typically difficult to get started writing good user stories
  - Here are 4 steps to make it easier

1. As a [role], I can [feature] so that [reason]
2. Use index cards and a sharpie
3. Make it testable with acceptance criteria or demo plan
4. Connect the dots

From: http://codesqueeze.com/the-easy-way-to-writing-good-user-stories/

# As a [role], I can [feature] so that [reason]

- Role – a person; feature – something your project does; reason – a solution to a problem the person has
  - This is a pattern that is commonly used for stories

    *As a account owner, I can check my balance online so that I can access my daily balance 24 hours a day.*

- Variations
  - As a [role], I want [feature] because [reason]
  - As a [role], I can [feature]
  - As a [role], I can [feature] so that [reason]

# Use index cards and a sharpie

- Although there is software out there to help you with this
  - Jira, Trello, Pivotal tracker
- Physically writing out stories facilitates keeping the story clear, concise, and of the appropriate size
  - Keep them short and sweet and unambiguous
    - Goal is to aid communication, not overly detailed or long-winded
  - It also enables you to doodle/draw the outline of the user interface
- If it doesn't fit, break up the story into sub-stories

# Make it testable with acceptance test or demo

- If they are short and sweet and without detail, how do we know when they are “done”?

- Include an acceptance test (what to demo when done):

  Scenario 1: Title

  Given [context]

  And [some more context]…

  When [event]

  Then [outcome]

  And [another outcome]…

Example

Scenario 1: Account balance is negative
Given the account’s balance is below 0
And there is not a scheduled direct deposit that day
When the account owner attempts to withdraw money
Then the bank will deny it
And send the account owner a nasty letter.

- All tests should fit on back of story card (in sharpie)
  - If they don’t, break up the story into two
  - You should be able to ***code*** them in a few lines of code