

# CS 4530: Fundamentals of Software Engineering

## Module 2, Lesson 1

### User Stories

---

Rob Simmons

Khoury College of Computer Sciences

© 2025 Released under the [CC BY-SA](#) license

# What are user stories?

---

- ...a tool to keep our designs focused on the goal of meeting needs of people
- ...formulaic statements of this form:

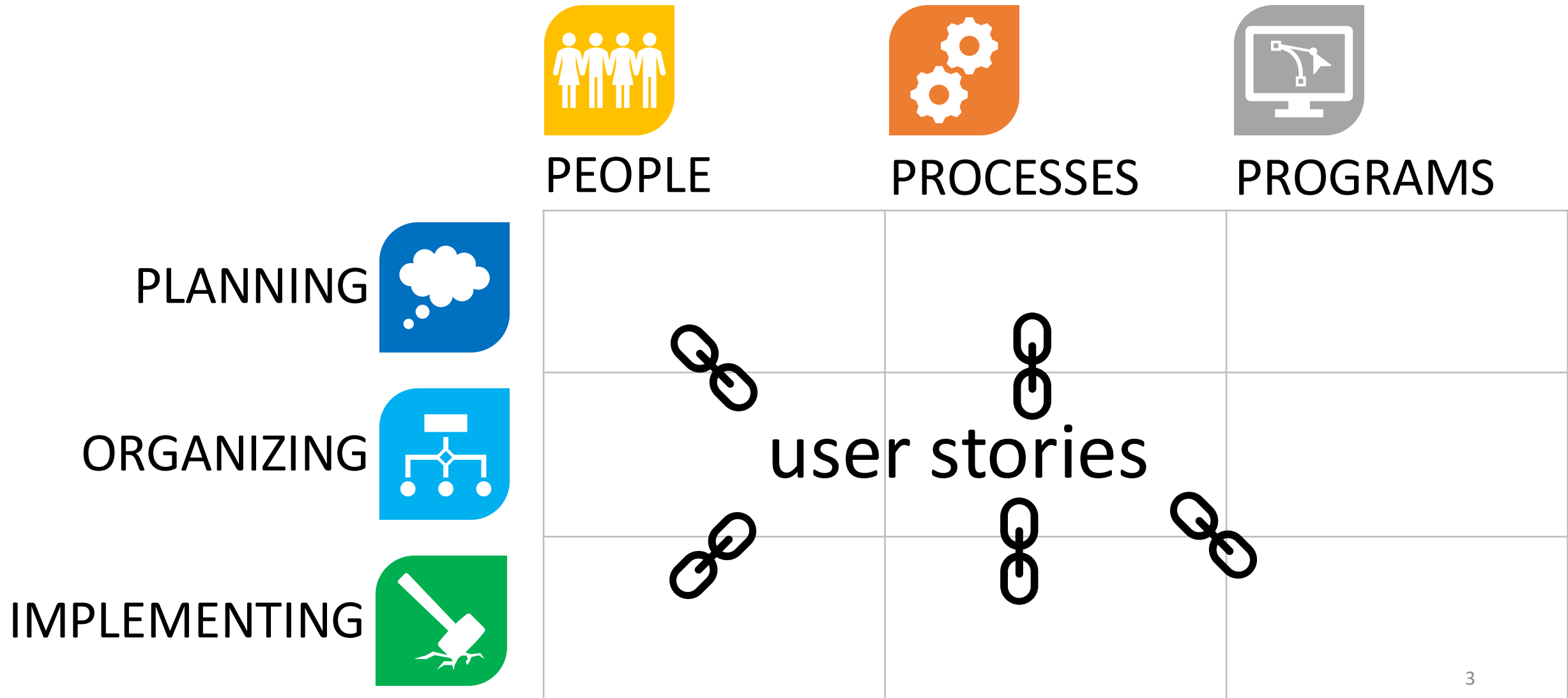
*As a <role>*

*I want <capability>*

*so that I can <get some benefit>*

- ...a tool for keeping large collaborative teams on the same page

# Where do user stories fit in?



# Learning Goals for this Lesson

---

- At the end of this lesson, you should be able to
  - Explain the structure of a user story
  - Identify and fix user stories that don't have the correct structure
  - Define the relationship between conditions of satisfaction and user stories, and the difference between essential, desired, and extension conditions
  - Identify whether a given condition of satisfaction actually relates to a given user story
  - Propose new conditions of satisfaction for new user stories

# Roles: “who”

---

- Roles are positions or functions that people inhabit!
  - “As a web server...” is not a user story!
  - “As a human being...” is better, but that’s still not a role
  - “As a user...” is almost always a cop-out
- The person is not you!

*As a <role>  
I want <capability>  
so that I can <get some benefit>*

# Capabilities: “what”

---

- A capability is a specific benefit
  - Because we’re building software, this is usually a benefit we can provide by software, otherwise let’s not build software
- A capability relates to a role
  - “As a teacher at northeastern, I want to be able to access a laser cutter so I can finish an art project” is not a good user story.

*As a <role>  
I want <capability>  
so that I can <get some benefit>*

# Capabilities: “what”

---

- A capability is not a product
  - “As a College Administrator, I want a web application that does <this> and <that> so that I can...” is not a user story!
  - This is easy wrong in practice! Sometimes you really want to build a tic-tac-toe game.
  - You’re supposed to ask “do I even need to build this?”

*As a <role>  
I want <capability>  
so that I can <get some benefit>*

# Benefit: “why?”

---

- Benefits are key for user stories actually focusing on what matters to the user.
- If a specific feature doesn't relate to the benefit...
  - ...maybe that feature isn't worth building.
  - ...maybe that feature is part of a different user story.  
(Maybe we should prioritize that different user story instead?)

*As a <role>  
I want <capability>  
so that I can <get some benefit>*



# Examples:

---

- As a College Administrator, I want to keep track of students, the courses they have taken, and the grades they received in those courses, so that I can advise them on their studies.
- As a driver, I want to be able to report potholes to the city so that the town can more quickly act to keep me safe.
- As a pedestrian, I want to be able to report potholes to the city so that drivers stop dangerously swerving onto the sidewalk when I'm walking.
- As a card game enthusiast, I want to be able to play blackjack online so that I can...

# Conditions of Satisfaction fill in details

---

- Each condition of satisfaction
  - Describes a testable behavior, from the user's point of view
  - Must have a priority
  - Should be numbered within its user story



# Priorities

---

- **Essential (E)** means the project is useless without it.
- **Desirable (D)** means the project is less usable without it, but is still usable.
- **Extension (X)** describes a user story or COS that is may not be achievable within the scope of the project. These might be things you'd want "in the next version".



# Worked Example: Pothole reporting system

A town is designing a system where citizens can report potholes and the town can monitor progress on repairing them.



# User Story #1

---

- As a car commuter, I want to be able to report potholes to the city so that the town can more quickly act to keep me safe.



# Conditions of Satisfaction

---

- 1.1 I should be able to report the location of a pothole to the system (E)
- 1.2 I should be able to see whether the pothole I report has been repaired (E)
- 1.3 I should be able to see whether others have reported potholes near me (D)
- 1.4 I should be able to see an estimated time when the pothole should be repaired (X)

## User Story #2

---

- As a pothole-repair-truck driver, I want the system to display the potholes I should be working on today. (E)

# Conditions of Satisfaction

---

- 2.1 I should be able to see my list of potholes for today (E)
- 2.2 I should be able to report that I repaired a given pothole (E)
- 2.3 I should be able to report that I was unable to repair a given pothole, and to supply a reason (E)
- 2.4 My daily list of potholes should be listed in an order that cuts down the time I spend driving from job to job (D)



## User Story #3

---

- As a maintenance supervisor, I want to be able to control the order in which potholes are repaired (D?)

# Conditions of Satisfaction

---

- 3.1 I should be able to give a higher priority to potholes on a particular street (E)
- 3.2 I should be able to give a higher priority potholes in a particular neighborhood (E)
- 3.3 I should be able to see on a map where there are a lot of potholes (D)
- 3.4 I should be able to see on a map which potholes that have been reported multiple times (D)

# Review

---

- It's the end of this lesson, so you should be able to
  - Explain the structure of a user story
  - Identify and fix user stories that don't have the correct structure
  - Define the relationship between conditions of satisfaction and user stories, and the difference between essential, desired, and extension conditions
  - Identify whether a given condition of satisfaction actually relates to a given user story
  - Propose new conditions of satisfaction for new user stories