ECE444: Software Engineering

Requirements 3: Documentation, Use Cases, User Stories

Shurui Zhou

The Edward S. Rogers Sr. Department
of Electrical & Computer Engineering
UNIVERSITY OF TORONTO

Learning Goals

- Understand tradeoffs of different documentation strategies
- Document requirements using use cases and user stories
- Evaluate the quality of a user story by INVEST
- (Understand risk and its role in requirements, specifically how it can be identified, analyzed, and then mitigated/handled in system design.)

Personas

"Personas are detailed descriptions of imaginary people constructed out of well-understood, highly specified data about real people"

—John Pruitt & Tamara Adlin

Partitioning the stakeholders into personas

Diversify your selections

- The common case (most users)
- The extremes (rare, but demanding users)

Creating Personas

Identify important categories of stakeholder

- Roles describe the kind of work people do, or their relationship in time to the product
- Goals describe what the users hope to achieve
- Segments describe shared demographic, attitudes or behaviors of your users

User Roles and Goals

How to describe a role?

- Defined by tasks, job descriptions, responsibilities
- Occupation (shopper, assistant, manager)
- Sub-divide by status: new shopper, repeat customer

What do they care about? How do they feel?

- Defined by their goals
- Behavior (“only browsing”, “get it done”, “max sales”)
- Life phases (adolescence, parenthood, retirement)

User Segments

- Can we segment our users by demographics?
  - Age ranges
  - Gender
  - Income level
- What about attitudes or behaviors?
  - Physically active, always moving, can't slow down
  - Likes routine, avoids uncertainty, rigid
  - Telecommuter, works from home, free spirit
  - Experienced, technically minded, geek

Example Persona

Cher

Gender : Female

Age: 35

Status: Married, one kid

Job: Full time employee
Part-time distance learning student

Behavior & Belief

Cher is 35 years old, married to Luke and is a mother of a 5 year old. She works as a business analyst in Chicago and is a distance learning student at University of Washington. She wakes up early to exercise, drops her kid to school, goes to work and studies while commuting and late in the evening. She uses multiple apps to balance her work and life. Even in her hectic schedule she sometimes manages to go for ice skating and dancing.

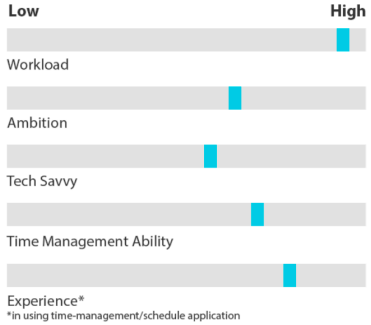
Goals

- To do well academically and advance in professional career
- No compromises on her family's well-being
- Continue to follow her hobbies
- To get good grades by finishing work on

Values

- Provide feedback on what could be done better
- Would prefer using only one app which fulfills her needs
- Distraction free

Characteristics



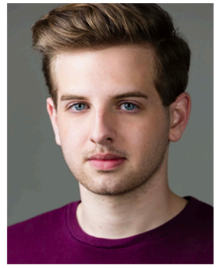
Fears

- Get delayed in important events such as picking up her kid from school

Pain Point

- Too many apps with redundant features

"Both family and work are very important. Plus, I also want to be a good student."



Tom

Gender : Male

Age: 25

Status: In a relationship

Job: Graduate student & TA

Behavior & Belief

Tom lives in a rented apartment and commutes to college daily. He works as a T. A. and aims to get a good job, so that he can repay his student loan. He manages his tasks by writing down work in a calendar application. He stopped using time tracking software because he forgot to record his activities. He believes that quality of work is important and often spends huge amount of time doing one task. He mostly uses his laptop for his work and has an internet connection.

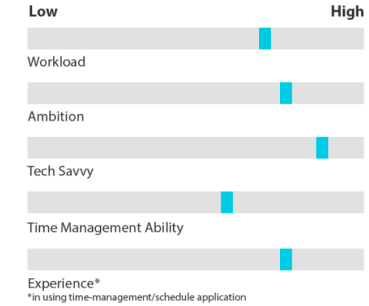
Goals

- To get a good job
- To repay his student loan
- To get good grades by finishing work on time

Values

- Automated tracking to record activities
- Analysis reports to better split time

Characteristics



Fears

- Forget to record his tasks

Pain Point

- Hates to manually enter time

"I would like to track my time but I often forget."

Lee

Gender : Male

Age: 23

Status: Single

Job: Graduate student, first year

Behavior & Belief

Lee had just started his first year in the MSE program. He is taking 51 units and is still getting the feel of how the workload is. But so far, he has been overwhelmed. He has had a lot of sleepless nights. He used the to-do list app on his phone but has not been using it since, Canvas has that feature. He has trouble focusing on a task for more than 30 minutes, so he does it over a period of time. Apart from his study, he practices vocal singing and plays badminton every day. He also likes to cook different dishes. He's on his mobile phone almost all the time to access the social media.

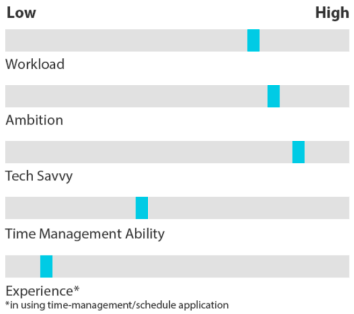
Goals

- Get a high GPA without burning out
- To continue following his hobbies

Values

- Get a reminder when he is behind schedule
- Get motivation to work
- Free to use

Characteristics



Fears

- Missing a deadline

Pain Point

- Managing deadlines

"My heart is in the work. It's hard to balance."

Partitioning the stakeholders into personas

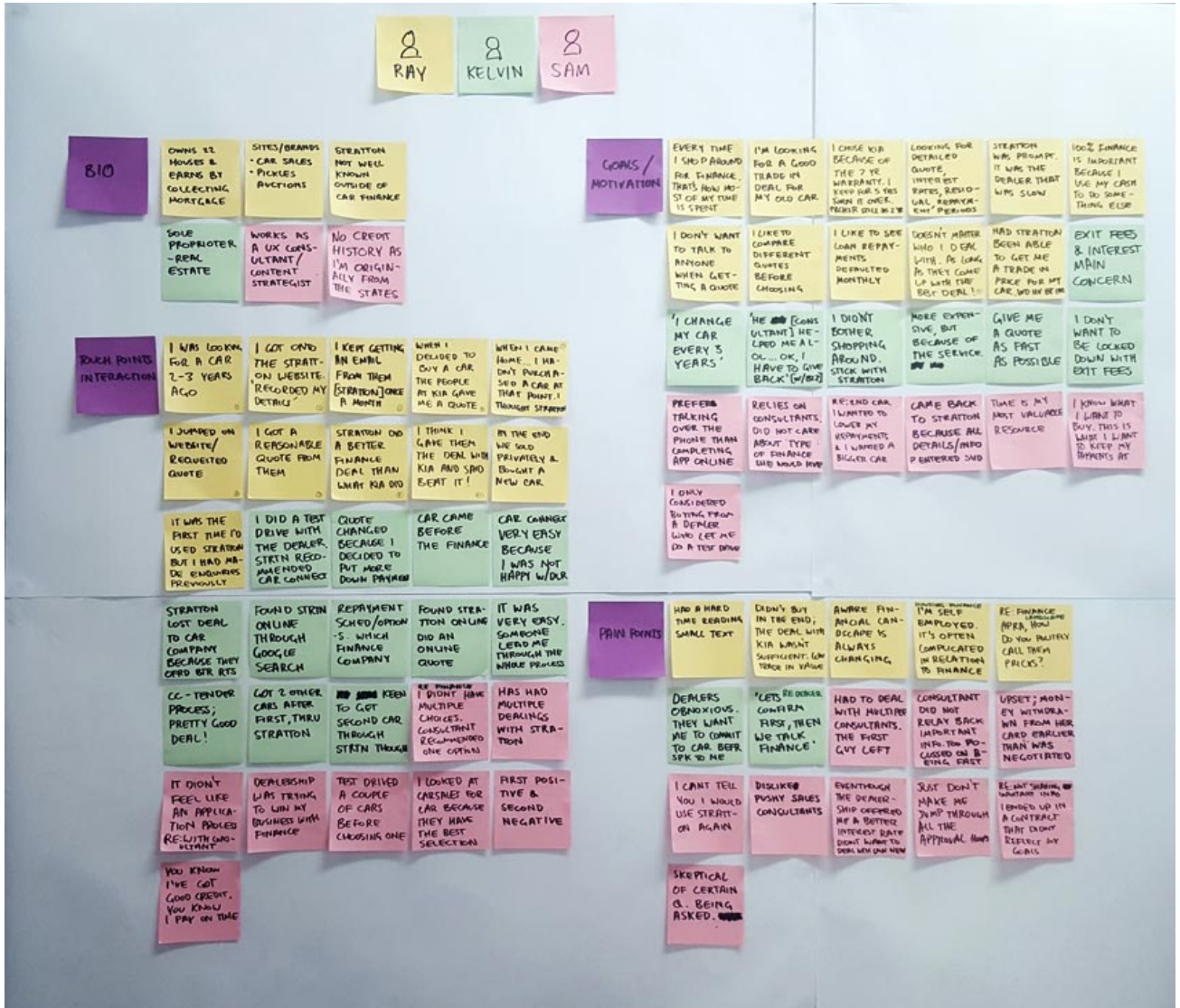
- Use a data-driven approach, whenever possible
  - Data collected using surveys or focus groups
  - Data reported in research studies
  - Data inferred using affinity diagrams
- Diversify your selections
  - The common case (most users)
  - The extremes (rare, but demanding users)

Synthesis Interviews through Affinity Diagrams

Synthesis Interviews through Affinity Diagrams

Synthesis Interviews through Affinity Diagrams

- biographical information
- frustrations
- interactions/touch points
- goals/motivations



Documentation Requirements

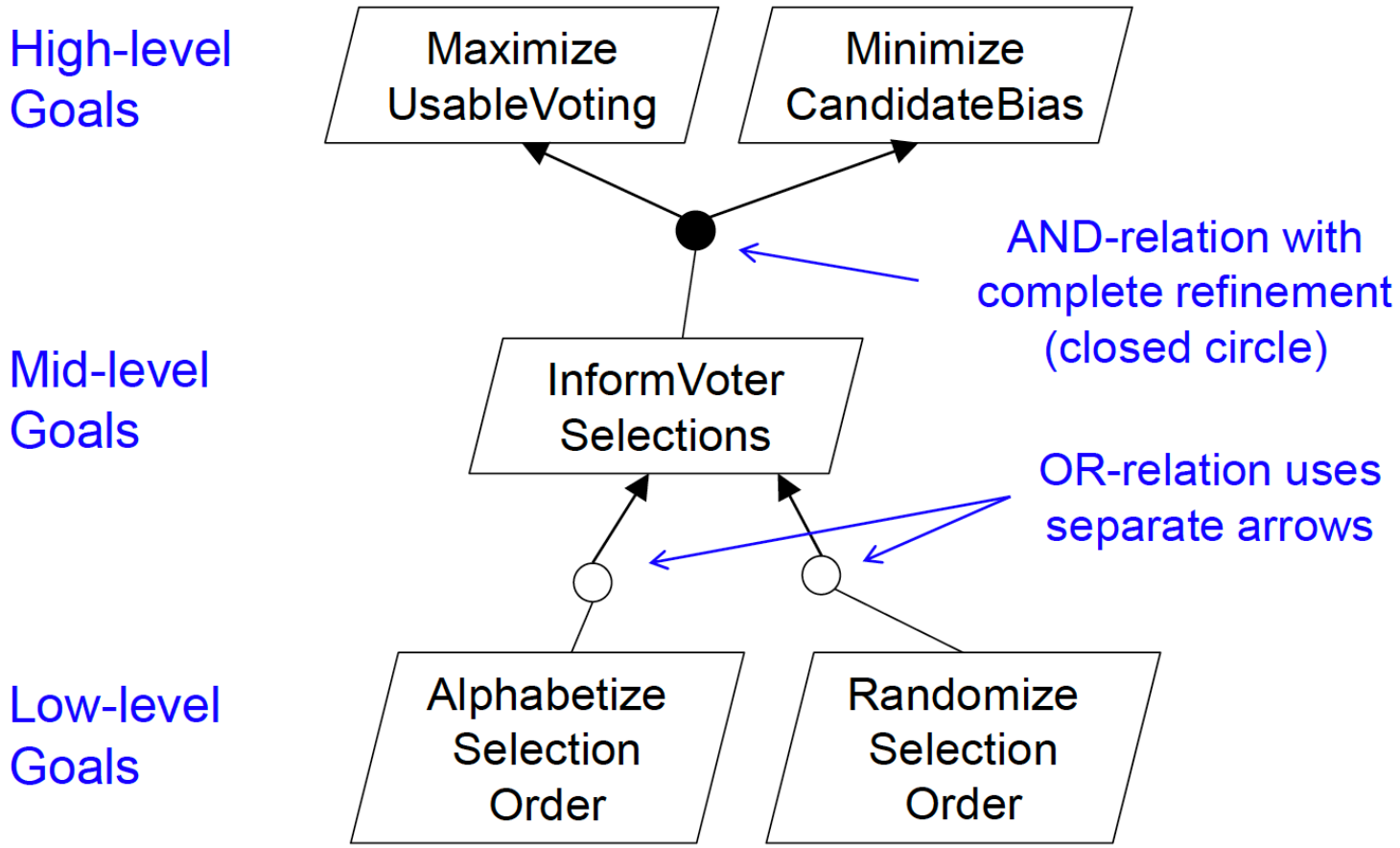
Goals

Begin convergence

Goal

- What is the purpose of this project?
- What are the problems it will solve?
- What improvements does your product offer over current solutions?
- What is the product vision?

Goal Diagrams



Goals - example

- Our goal is to create a mobile version of the website. Sometimes users click on a link in an email notification using their mobile phone and need to be able to access our application from mobile Chrome or Safari.
- We want to meet feature parity with most functions - except we can skip creating events.

Example2:

For the case of building a ToDo app, our primary purpose is creating an app that lets users track and mark off their daily tasks and important commitments. It will help them stay organized and ensure they don't overlook any items, without requiring much interaction from the user. The app will need to perform well such that it stands out from its existing competitors, and ultimately act as a "better mousetrap" in terms of its usability and functionality.

Many different forms

- Informal vs formal
- Unstructured vs structured
- Text vs diagrams
- Structured text common in practice
- Tool supported for traceability and process integration

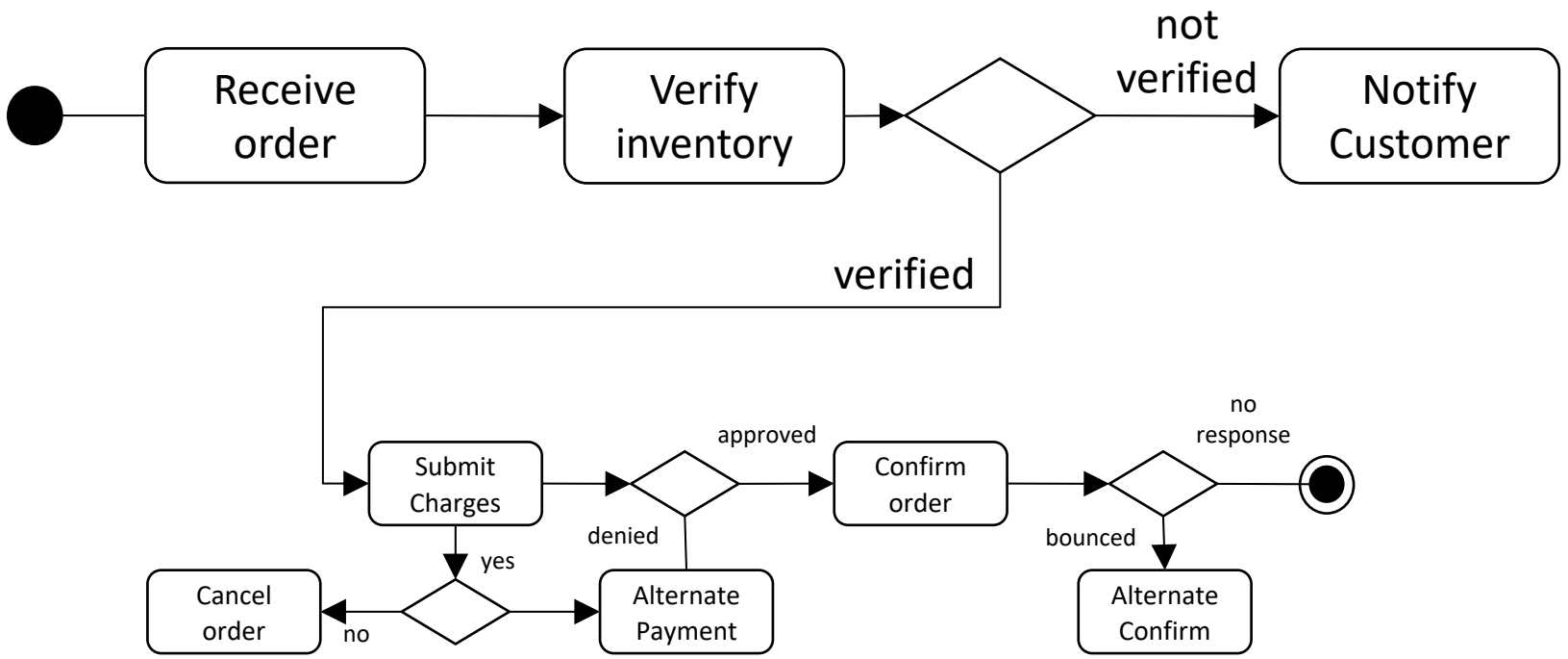
Software Requirements Specification (SRS)

- Formal requirements document
- Several standards exists
- Often basis for contracts

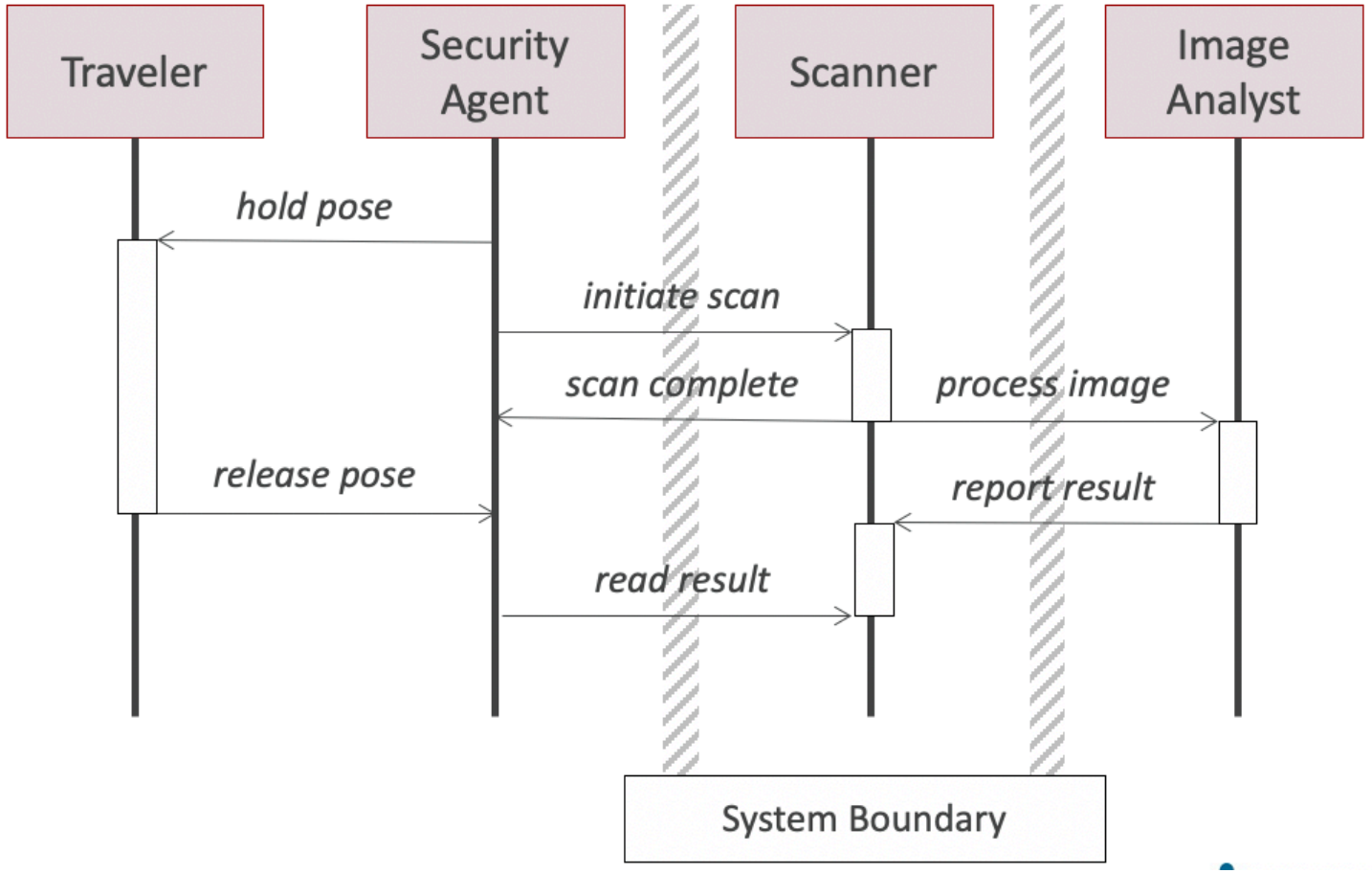
Table of Contents

Activity Diagrams

- Activity diagrams (or flow charts) represent the logic in a graph notation



Sequence Diagramming



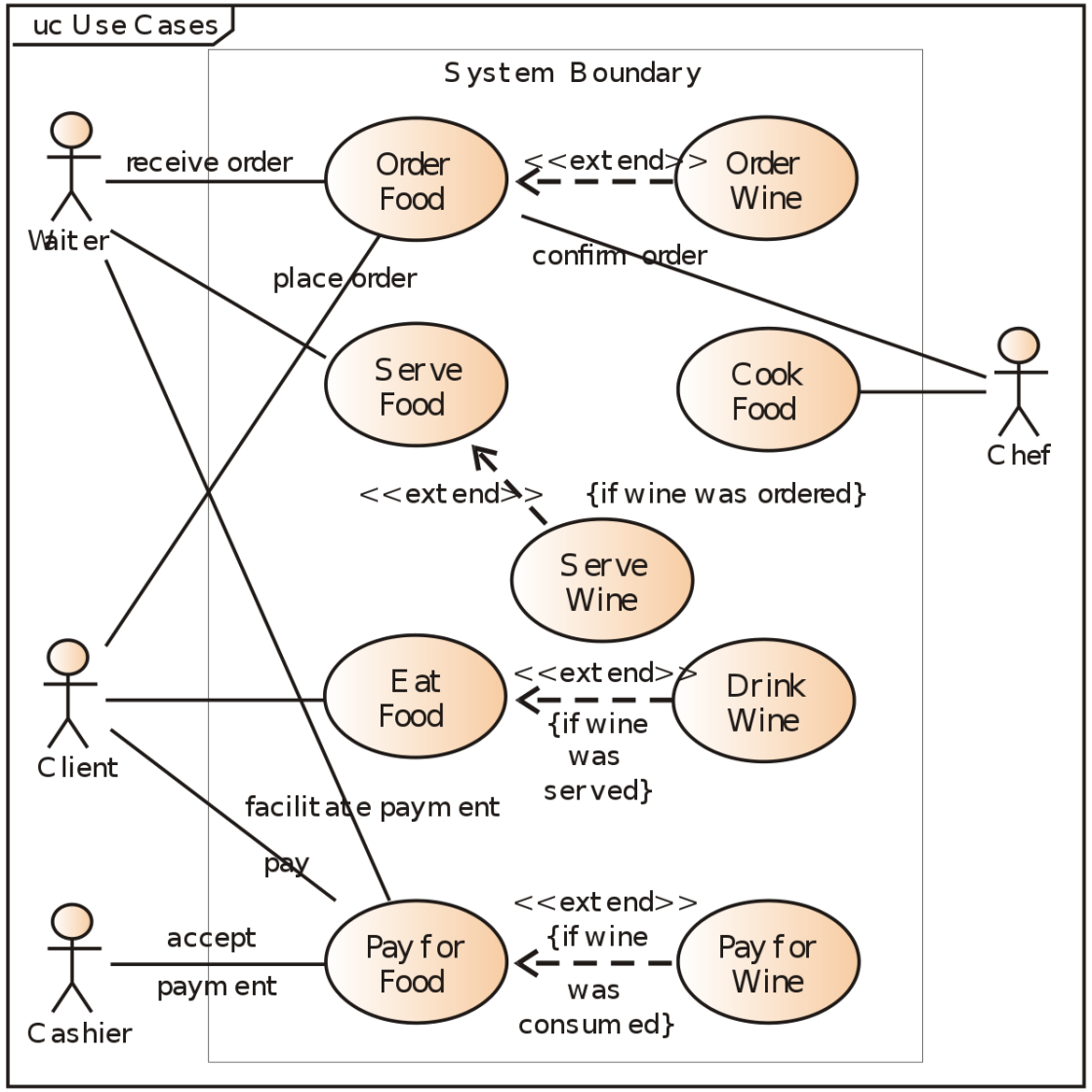
Formal Specification

- Logical expressions of shared actions at the interface of the machine
- Includes linking domain properties and agent actions as pre- and post-conditions

$$\forall\, s\, \forall\, c(\text{enrolled}(s, c) \Rightarrow \text{student}(s) \wedge \text{course}(c))$$

Use Case Diagram

- Actor + action



Use Case

Use Cases help requirements analysts to…

- Identify actors and events around the system
- Define the system boundary –what is or is not within the system scope?
- Investigate early design interactions

(uses cases need not be descriptions of the final design)

Defining actors/agents

- An actor is an entity that interacts with the system for the purpose of completing an event [Jacobson, 1992].
  - Not as broad as stakeholders.
- Actors can be a user, an organization, a device, or an external system.

Example: Place an order?

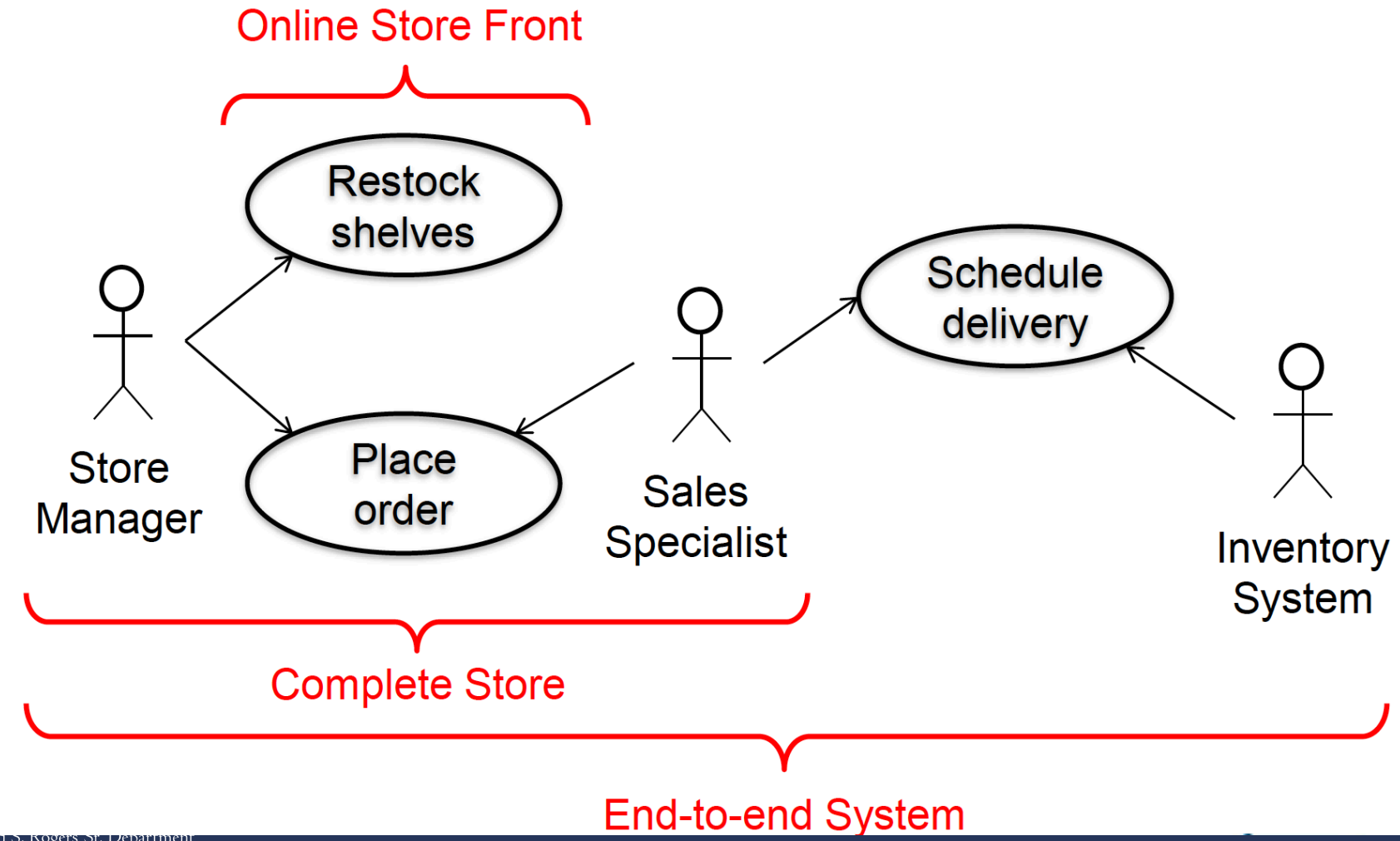
Buyer's View

- Selecting the products
- Reviewing the order
- Submitting the order
- Receiving delivery

Seller's View

- Receiving the order
- Checking the inventory
- Filling the order
- Shipping the order
- Confirming delivery

Defining the system boundary



Pre-and post-conditions

- Pre-conditions: true before the use case begins
- Post-conditions: true at the end of the use case
- Should be written at the same “level of detail” as the use case
- Apply to the state of the system, not the environment outside the system [Armour & Miller]
  - The book has a status of borrowed
  - The patron is free to leave the library with the book

Use Case Templates

Use Case Name	Place order
Actors	(Primary) Store Manager, Sales Specialist
Pre-conditions	
Flow of events	
Post-conditions	

Use Case Templates

Use Case Name	Place order
Actors	(Primary) Store Manager, Sales Specialist
Pre-conditions	The store manager is under-stocked or the manager anticipates an increase in next period's sales
Flow of events	
Post-conditions	An order to restock the shelves is being processed

Use Case Templates

Use Case Name	**Place order**
Actors	(Primary) Store Manager, Sales Specialist
Pre-conditions	The store manager is under-stocked or the manager anticipates an increase in next period's sales
Flow of events	1. Sales specialist identifies manager's account 2. Manager finds the products to reorder 2.1 Manager browses or searches by keyword 2.2 Manager decides product quantities 3. Manager reviews and places the order 4. Specialist receives and processes the order
Post-conditions	An order to restock the shelves is being processed

Surfacing Assumptions

- The Manager has an Internet connection
- The System manages user accounts
- The Manager has a list of products that they can provide by browsing and searching

Use Case Name	**Place order**
Actors	(Primary) Store Manager, Sales Specialist
Pre-conditions	The store manager is under-stocked or the manager anticipates an increase in next period's sales
Flow of events	1. Sales specialist identifies manager's account 2. Manager finds the products to reorder 2.1 Manager browses or searches by keyword 2.2 Manager decides product quantities 3. Manager reviews and places the order 4. Specialist receives and processes the order
Post-conditions	An order to restock the shelves is being processed

Identify Key system behaviors

- What system activities must be performed to help fulfill the use case? (These may not be transparent to a user)

Use Case Name	**Process order**
Actors	Sales Specialist, Store Manager
Pre-conditions	The store manager places an order
Flow of events	1. Specialist receives the order 2. Specialist verifies inventory contains order 3. Specialist submits charges for payment 4. Specialist sends manager order confirmation
Post-conditions	Order is scheduled for fulfillment and shipping

Alternative Flows

Alternative flows include:

- Different processing options based on user input
- Decision taken within an existing flow
- An exception condition that occurs in a flow

Alternative Flows & Exceptions

Use Case Name	**Process order**
Actors	Sales Specialist, Store Manager
Pre-conditions	The store manager places an order
Flow of events	1. Specialist receives the order 2. Specialist verifies inventory contains order 3. Specialist submits charges for payment 4. Specialist sends manager order confirmation
Post-conditions	Order is scheduled for fulfillment and shipping
Alternate flows and exceptions	

Alternative Flows & Exceptions

Use Case Name	**Process order**
Actors	Sales Specialist, Store Manager
Pre-conditions	The store manager places an order
Flow of events	1. Specialist receives the order 2. Specialist verifies inventory contains order 3. Specialist submits charges for payment 4. Specialist sends manager order confirmation
Post-conditions	Order is scheduled for fulfillment and shipping
Alternate flows and exceptions	• The inventory does not contain an ordered item • The payment is not authorized • The payment service times out • The order confirmation is returned (bounces)

Alternative flow descriptions

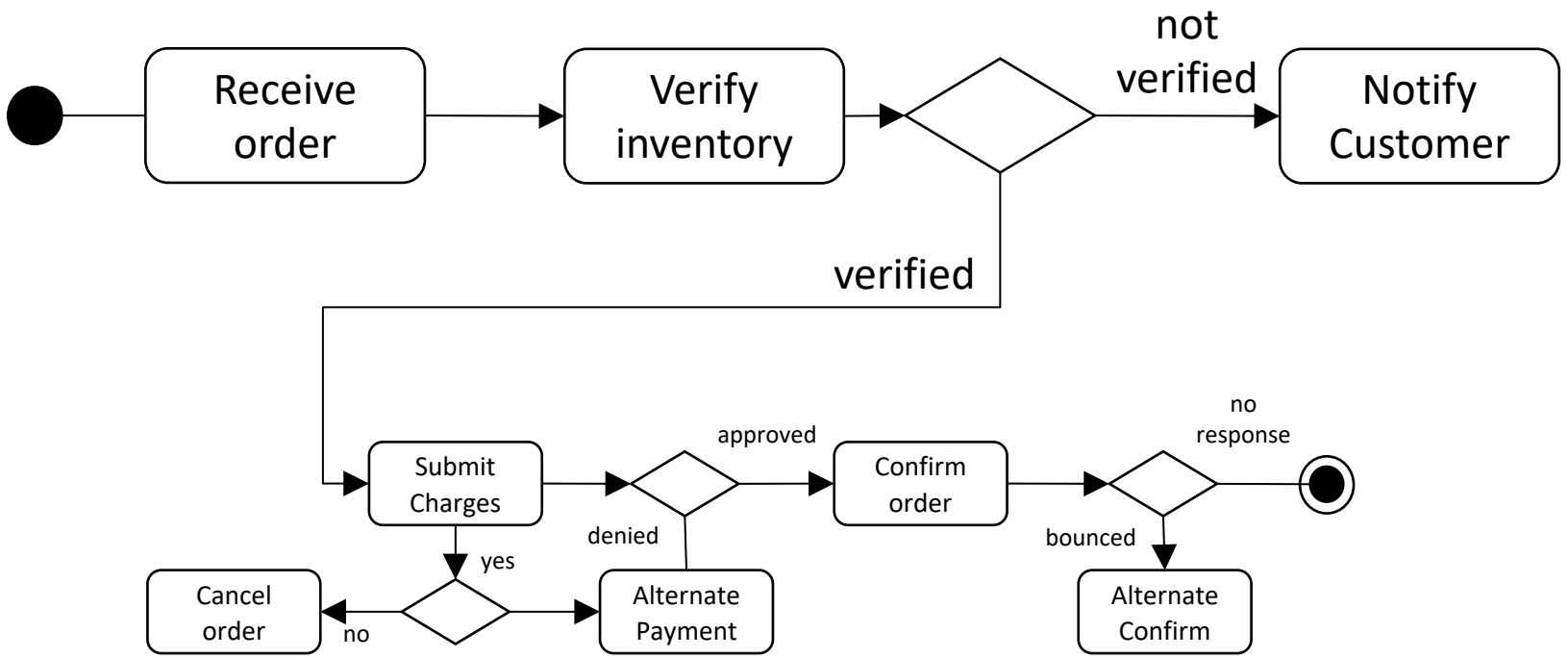
Alternative Name	**Unauthorized Payment**
Actors	Sales Specialist, Store Manager
Insertion Point:	Step 3, specialist submits charges for payment
Pre-conditions	The payment processing is not authorized
Flow of events	1. Specialist sends a problem notice to the store manager 2. Store manager may submit an alternative payment method
Post-conditions	A new payment method is submitted, repeat Step 3, or the order is cancelled
Non-behavioral requirements	The notice provides a convenient method to submit an alternative method of payment

Integrating conditional logic

Use Case Name	**Process order – Integrated logic**
Actors	Sales Specialist, Store Manager
Pre-conditions	The store manager places an order
Flow of events	1. Specialist receives the order 2. Specialist verifies inventory contains order **If** the inventory does not contain the order… 3. Specialist submits charges for payment **If** the payment is not authorized… 4. Specialist sends manager order confirmation **If** the confirmation is returned…
Post-conditions	Order is scheduled for fulfillment and shipping

Activity Diagrams

- Activity diagrams (or flow charts) represent the logic in a graph notation



Non-behavioral Requirements

- Performance – How long will the use case take to complete? What are normal and peak conditions?
- Capacity – How many actor instances must be supported?
- Security – Are there confidentiality, integrity or availability concerns?
- Usability– What do actors need to do to fulfill the use case?

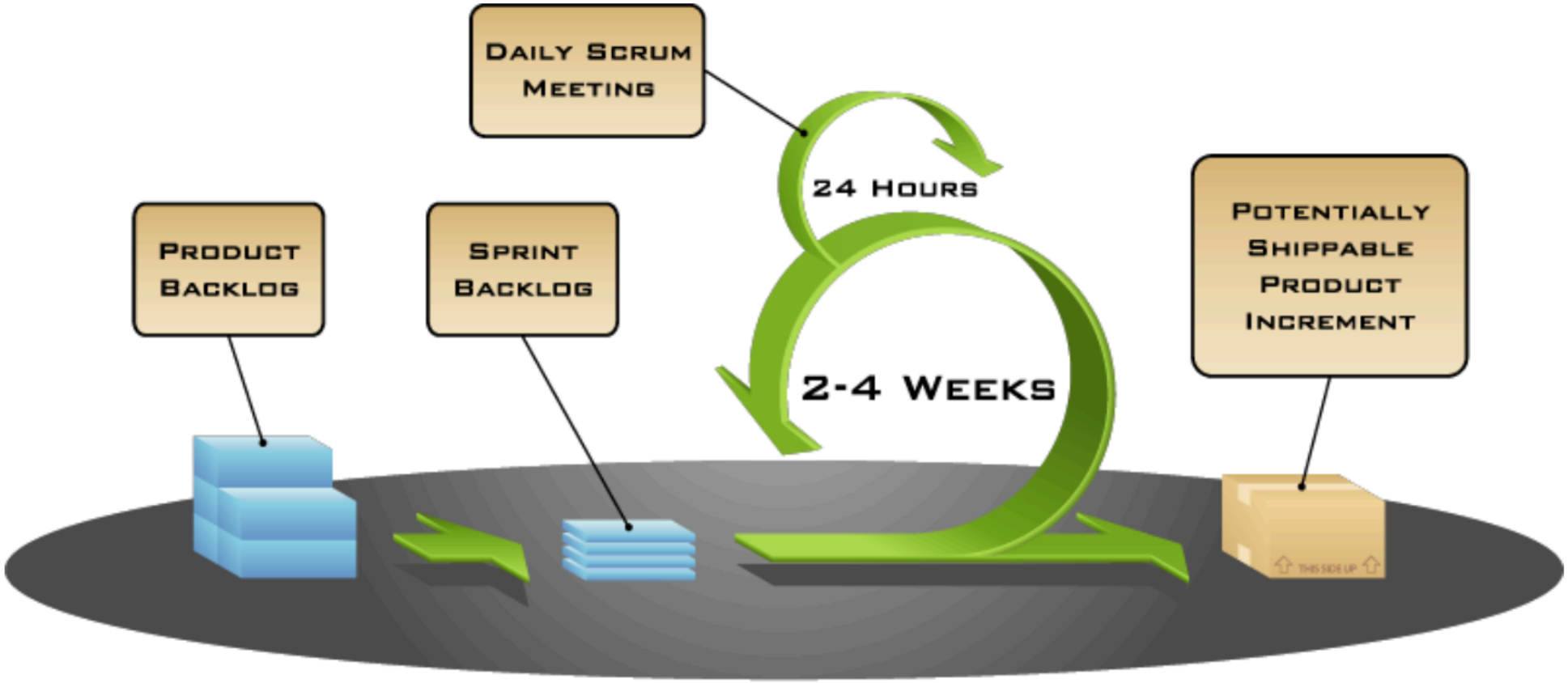
Non-behavioral Requirements

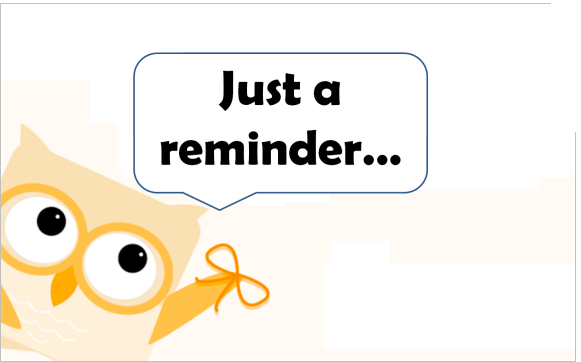
Use Case Name	**Process order**
Actors	Sales Specialist, Store Manager
Pre-conditions	The store manager places an order
Flow of events	1. Specialist receives the order 2. Specialist verifies inventory contains order 3. Specialist submits charges for payment 4. Specialist sends manager order confirmation
Post-conditions	Order is scheduled for fulfillment and shipping
Non-behavioral requirements	• Inventory is routinely refreshed and kept up to date • Orders should be processed within 10 minutes

Use cases Template

Use Case Name	(Title)
Scope	System under design
Level	User level, subprocess level
Primary actor	(actors can be primary, supporting, or offstage)
Stakeholders, interests	Important! A use case should include everything necessary to satisfy the stakeholders' interests.
Preconditions	What must always be true before a scenario begins. Not tested; assumed. Don't fill with pointless noise.
Success guarantees.	Aka post conditions
Main success scenario	Basic flow, "happy path", typical flow. Defer all conditions to the extensions. Records steps: interaction between actors, a validation, a state change by the system.
Extensions	Aka alternate flows. Usually the majority of the text. Sometimes branches off into another use case.
Special requirements	Where the non-functional/quality requirements live.
Technology and data variations list	Unavoidable technology constraints; try to keep to I/O technologies.
Frequency of occurrence	
Miscellaneous	

Agile?


Daily Scrum Meeting
24 Hours
Product Backlog
Sprint Backlog
2-4 Weeks
Potentially Shippable Product Increment


Just a reminder...

User Stories

- Informal descriptions of user-valued features scheduled for implementation
- Details left for negotiation with customer later or pointer to real requirements
- Common agile development practice

User Stories

Who (User)

This should describe a fairly detailed user. It is not sufficient to just say "user." Strive towards something like "broke college student on a mobile device user." When we express the **who** with more detail we are able to better empathize with that particular user, determine the best solution and uncover implicit needs.

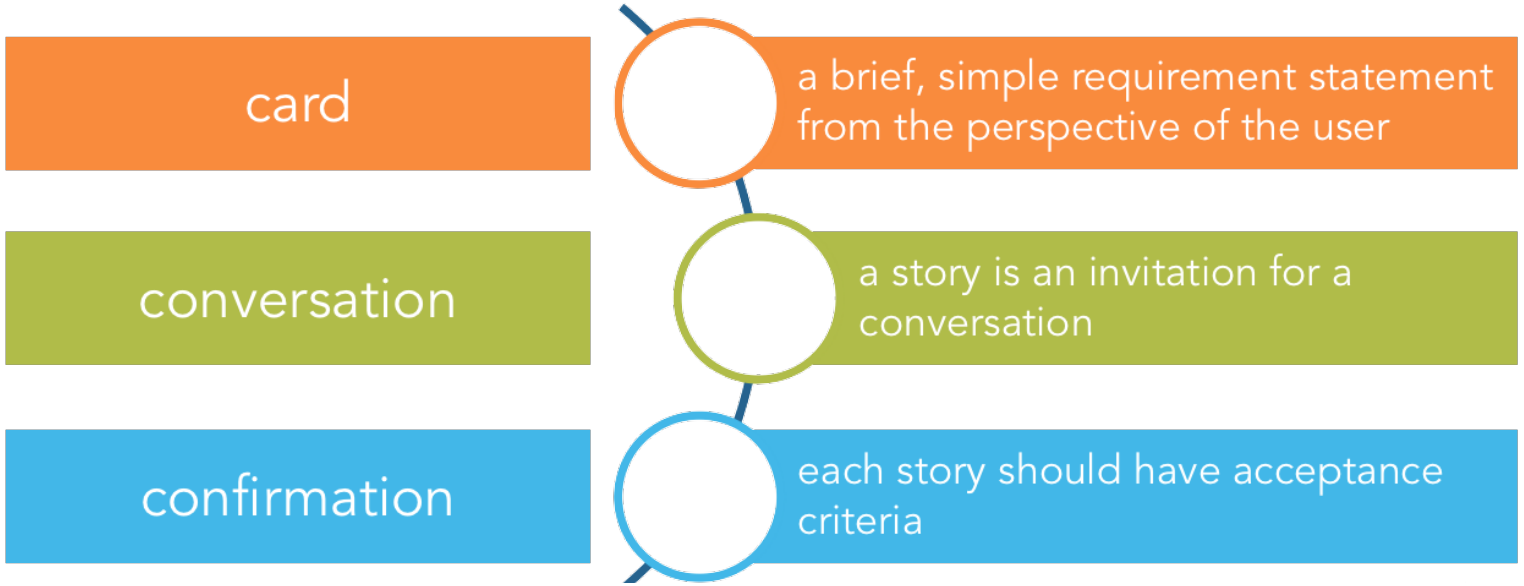
What (Goal)

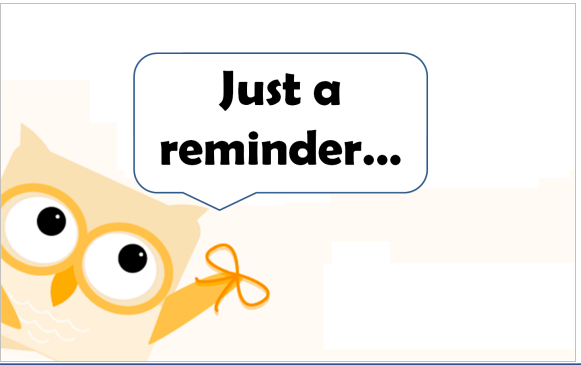
The goal or action the user intends to take.

Why (Benefit)

Expressing the benefit to the user is by far the most important in my experience. Some of the most creative and inexpensive solutions come from the developers and users understanding why they are building something.

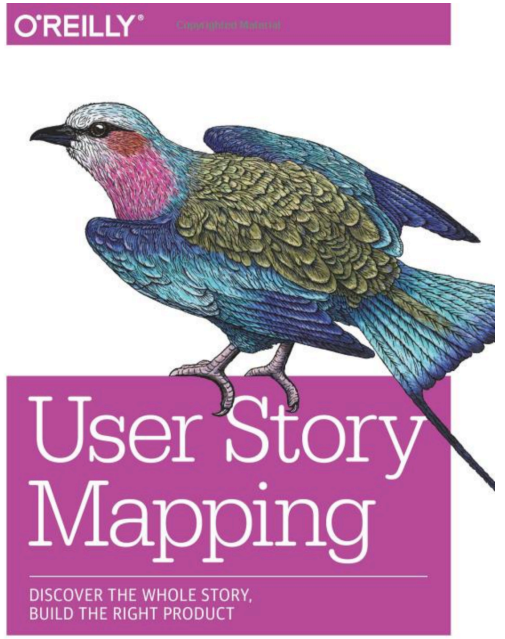
User Stories -- Concept of 3C's



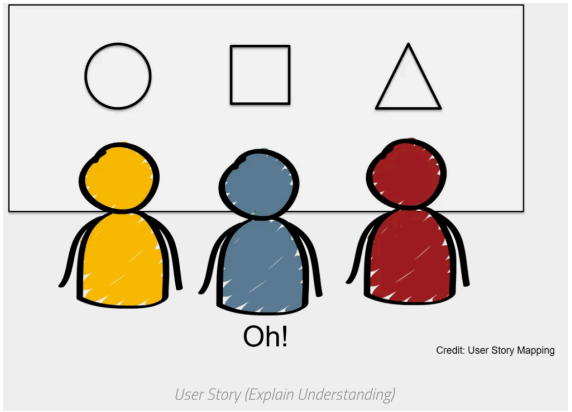


The conversation

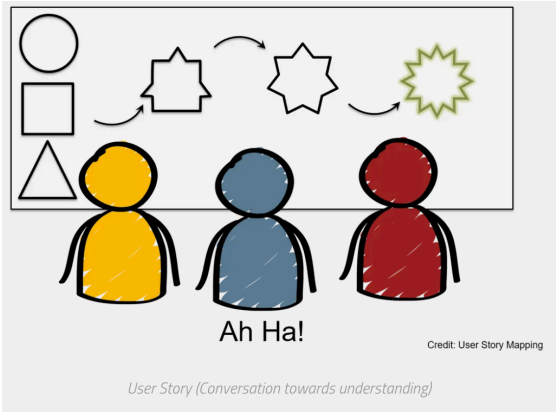
- An open dialog between everyone working on the project and the client
- Split up Epic Stories if needed



User Story (Not Shared Understanding)



User Story (Explain Understanding)



User Story (Conversation towards understanding)

User Story (Shared Understanding)

User Story Examples

- "As a [persona]": Who are we building this for? We're not just after a job title, we're after the persona of the person. Max. Our team should have a shared understanding of who Max is. We've hopefully interviewed plenty of Max's. We understand how that person works, how they think and what they feel. We have empathy for Max.
- "Wants to": Here we're describing their intent — not the features they use. What is it they're actually trying to achieve? This statement should be implementation free — if you're describing any part of the UI and not what the user goal is you're missing the point.
- "So that": how does their immediate desire to do something this fit into their bigger picture? What's the overall benefit they're trying to achieve? What is the big problem that needs solving?

User Story Examples

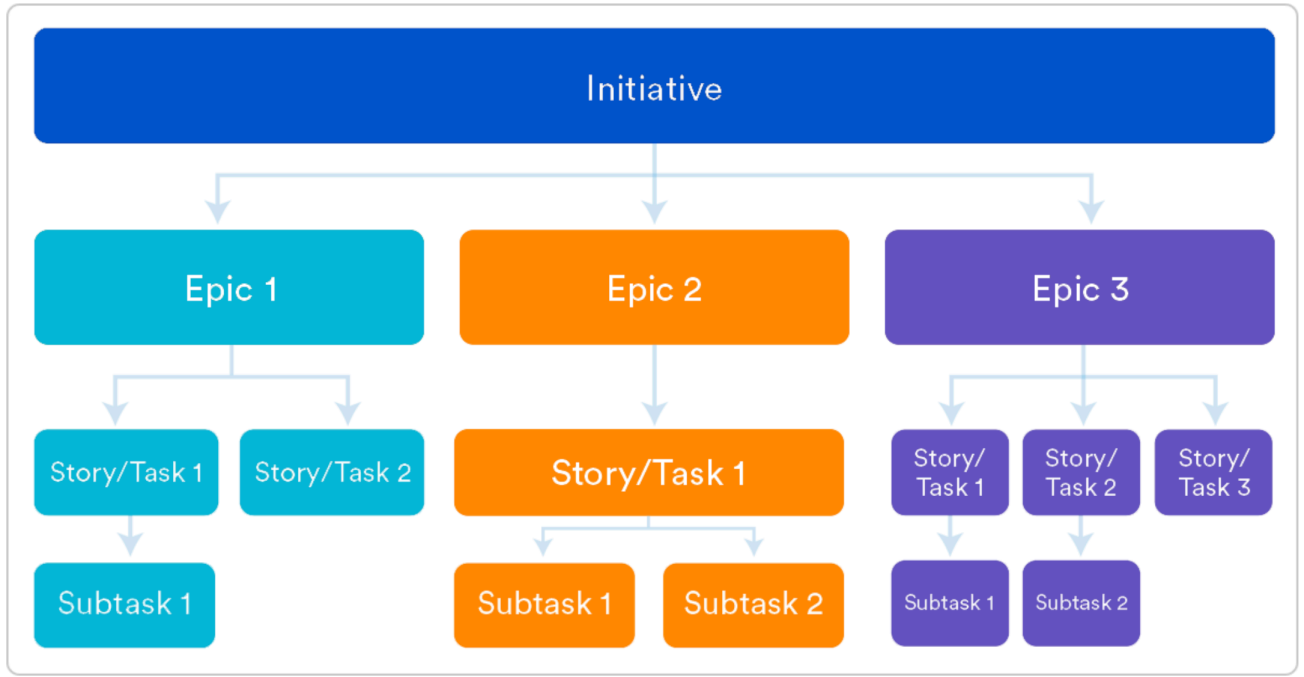
- iPhone users need access to a vertical view of the live feed when using the mobile app.
- Desktop users need a “view fullscreen” button in the lower right hand corner of the video player.
- Android users need to be linked to apple store.

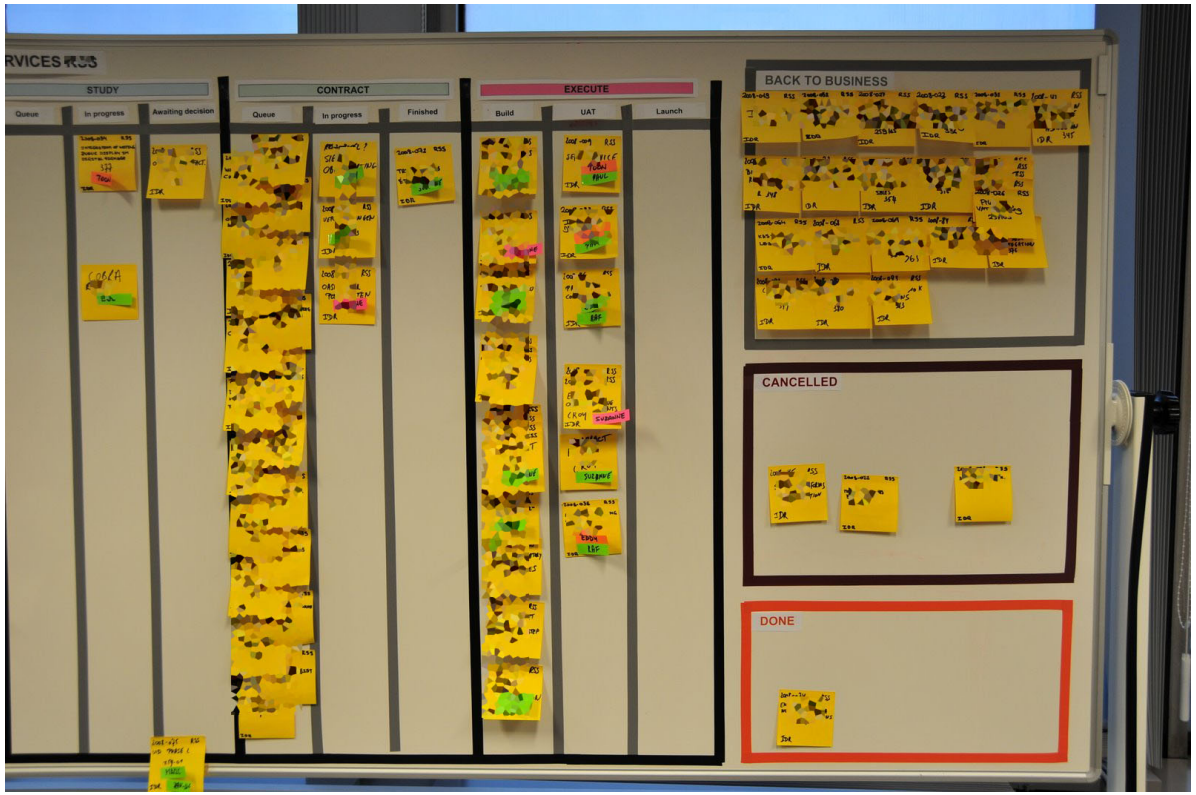
User Story Examples

- As Max, I want to invite my friends, so we can enjoy this service together.
- As Sascha, I want to organize my work, so I can feel more in control.
- As a manager, I want to be able to understand my colleagues progress, so I can better report our sucess and failures.

Use of User Stories

- Keep a board of user stories, group them into “epics”





The Confirmation

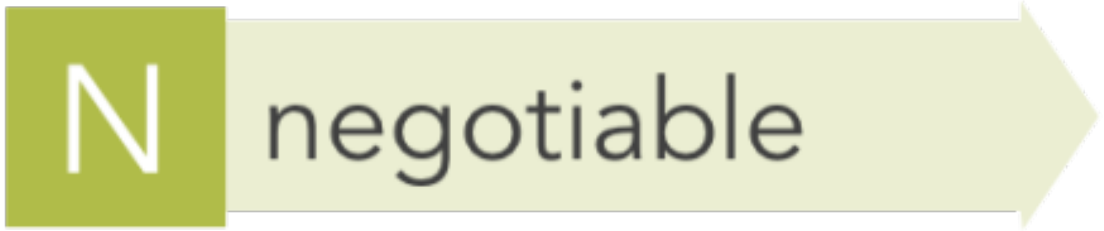
- A confirmation criteria that will show when the task is completed
- Could be automated or manual

How to evaluate user study?

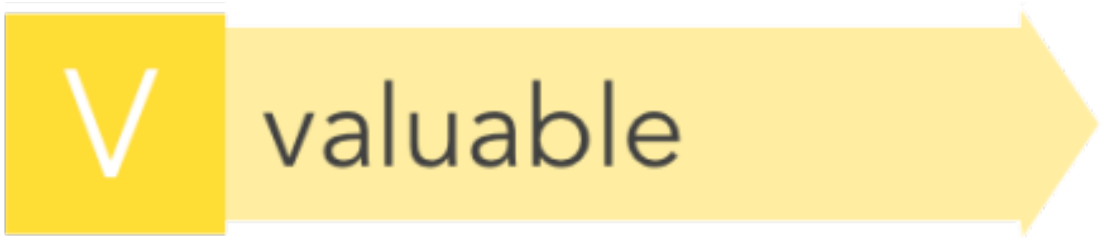
Follow the INVEST guidelines for good user stories!

one 80

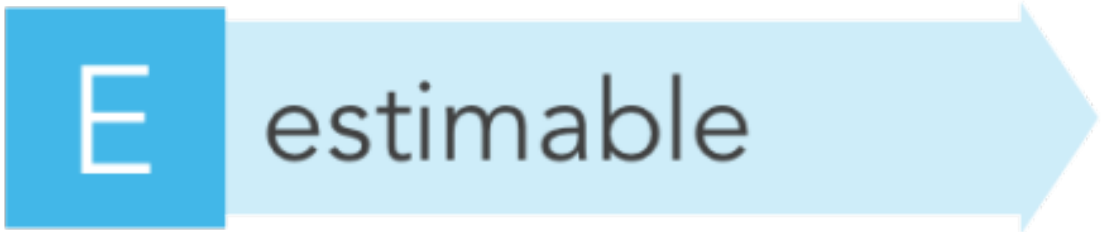
- Schedule in any order.
- Not overlapping in concept
- Not always possible



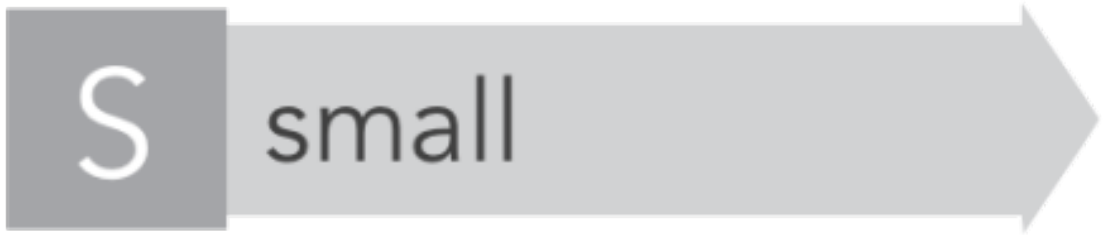
- Details to be negotiated during development
- Good Story captures the essence, not the details



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



- Helps keep the size small
- Ensure we negotiated correctly
- “Plans are nothing, planning is everything” -Dwight D. Eisenhower

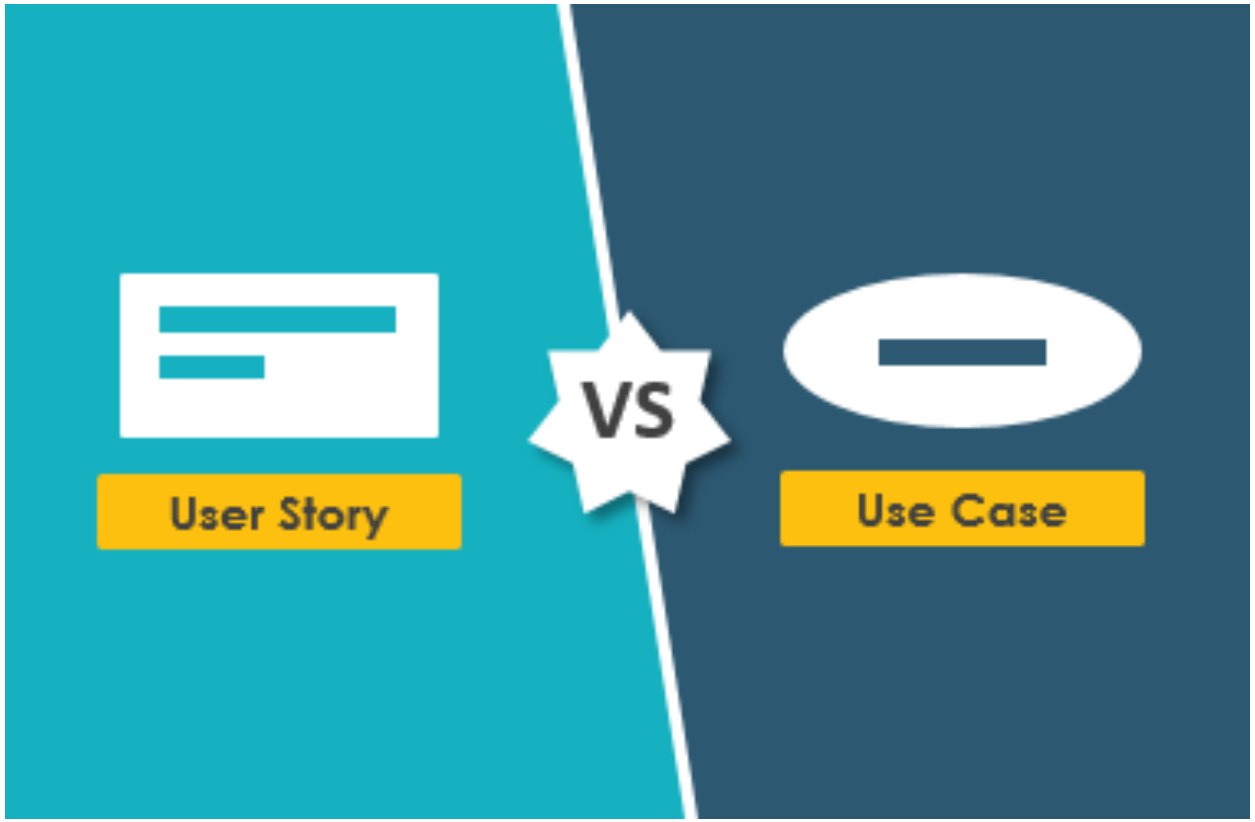


- Fit on 3x5 card
- At most two person-weeks of work
- Too big == unable to estimate

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

However...

"Is a User Story the same thing as a Use Case?"



Agile Development: User stories are the new requirements document

Is a User Story the same thing as a Use Case?

- **Not interchangeable**
- **User Stories** are centered on the result and the benefit of the thing you're describing
- **Use Cases** can be more granular, and describe how your system will act.

Use Cases vs User Story

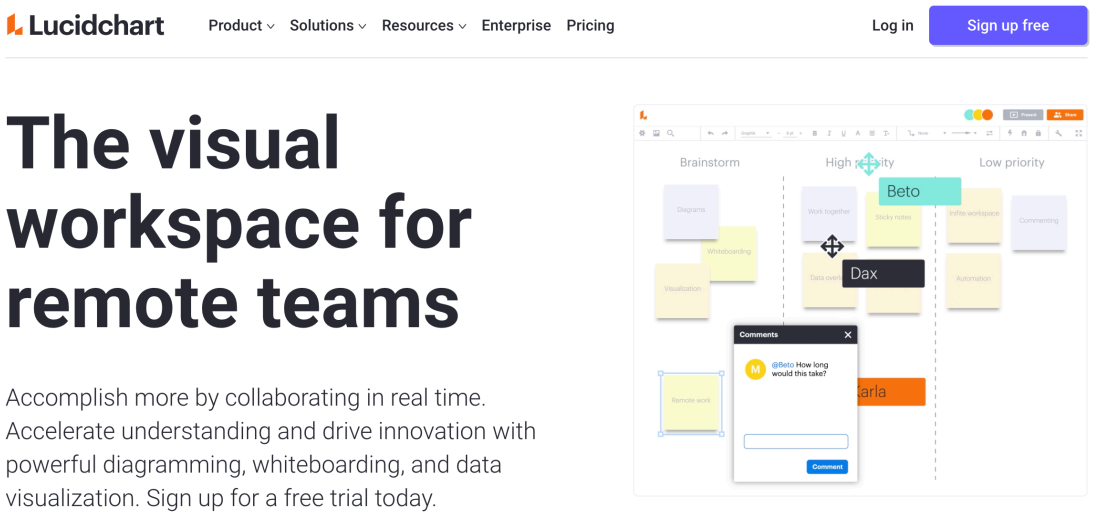
- Similarity
  - User Stories: user role, goal and acceptance criteria.
  - Use Cases: an actor, flow of events and post conditions
- Difference
  - Less details in User Story
  - Small increments for getting feedback more frequently, rather than having more detailed up-front requirement specification as in Use Cases.

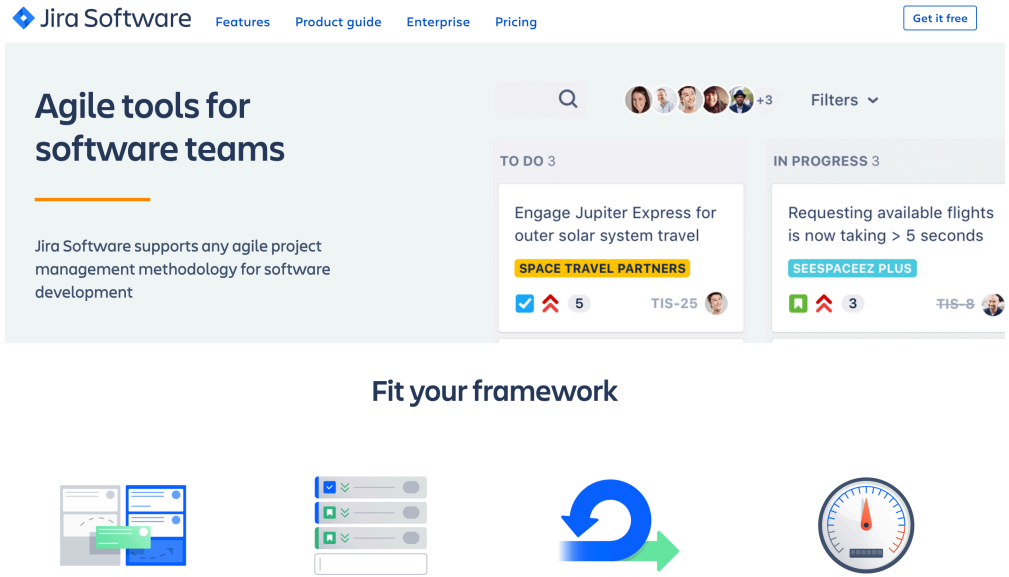
Why we still need Use Cases?

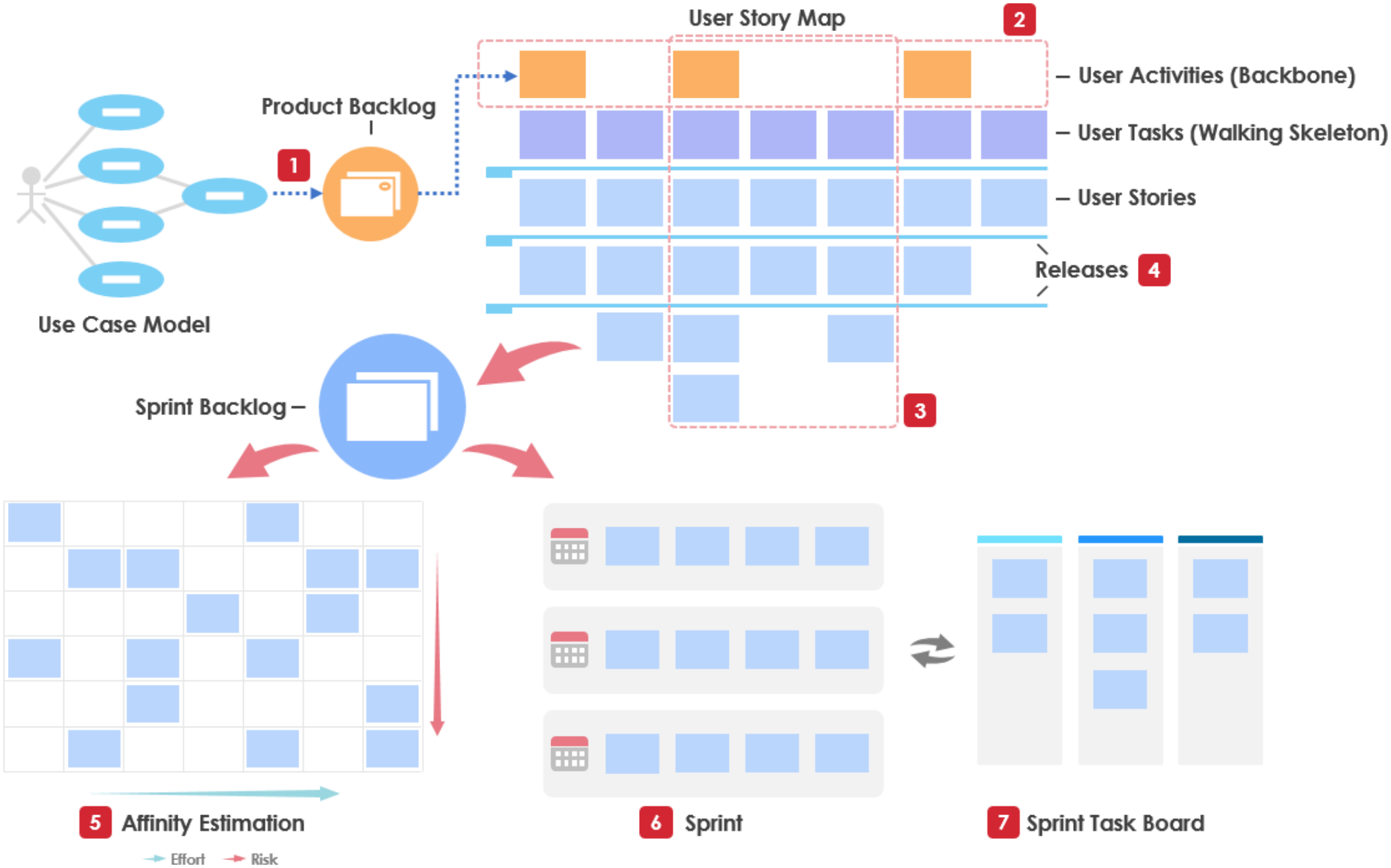
- Problem of User Story:
  - Lack of context
  - Sense of completeness that you covered all bases relating to a goal.
  - No mechanism for looking ahead at upcoming work.

Integrate Use Case, User Story and Story Mapping techniques

- Lucidchart
- Jira Agile
- Team Foundation Server
- BoardThing
- Stories on Board
- FeatureMap






User Story Map
2
– User Activities (Backbone)
– User Tasks (Walking Skeleton)
– User Stories
Releases
4
Product Backlog
1
Use Case Model
Sprint Backlog –
3
5
Affinity Estimation
Effort
Risk
6
Sprint
7
Sprint Task Board

Requirements prioritization

- Cost, time, and other limits
- Dependencies among requirements
- Nice to have
- Strategies to base on value contribution

Product Requirement Document (PRD)

1. Goals
2. User Personas
3. User Stories
4. Functional Requirements
5. Non-Functional Requirements
6. User interaction and design
7. Questions
8. Out of Scope

Summary

- Many documentation strategies; our focus is on *user stories*

Further Reading

- Larman, Craig. *Applying UML and Patterns: An Introduction to Object Oriented Analysis and Design and Interative Development*. Pearson, 2012. Chap. 6
- Van Lamsweerde A. Requirements engineering: From system goals to UML models to software. John Wiley & Sons; 2009. Chapter 2-4
- “Advanced Use Case Modeling, Volume I”, Frank Armour, Granville Miller,Addison-Wesley, 2001, Ch 8-10.
- https://aanimesh.files.wordpress.com/2013/09/applying-uml-and-patterns-3rd.pdf