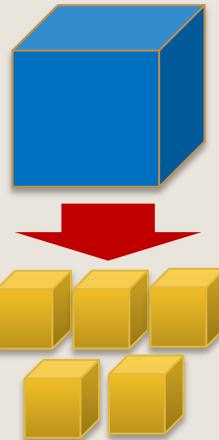
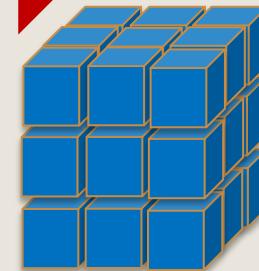


# Effective User Stories



Break down large intent into smaller, more manageable chunks

||| Deliver iteratively and incrementally  
Providing value with each iteration



## User Story Narrative

As a <Consumer of the value>  
I want to <Functionality (What)>  
so that <Value/benefit (Why)>

### Tips

- Write stories from the end user perspective (user can be a system)
- Understand **what** the user wants and **why** (not *how*)
- Don't assume just one kind of user of the system
- Don't confuse user stories with tasks. **Stories** describe what the user wants and why. **Tasks** are how we will achieve the goal/benefit of the story.

## Acceptance Criteria

Scenario 1: <summary of test>  
**Given** <precondition or assumption>  
And <additional precondition>  
**When** <perform an action>  
And <perform additional action>  
**Then** <this outcome will result>  
And <an additional outcome>

Acceptance Criteria are essentially a measure of whether a User Story has been correctly completed. A story may have multiple scenarios to define the acceptance criteria. They also help define the scope and create a shared understanding of the story.

## INVEST for Better Stories

**Independent:** Story can be valued, developed, and delivered on its own

**Negotiable:** It's a placeholder for requirements to be discussed, developed, and accepted

**Valuable:** Provides value to the user

**Estimable:** Small enough so effort to implement can be rapidly determined

**Small:** Small enough to fit into a single sprint but large enough to be valuable

**Testable:** Acceptance criteria is testable by Testers or Business Users

# User Story Example and Common Problems

## Example

As a customer service agent, I want to know the customer's benefits payment amount so that I can see that information and relay it to the customer.

### Acceptance Criteria

**Scenario 1:** Calculate benefits payment - Active

**Given** that a customer has a benefits plan on file associated with their account

**And** the benefits plan is active

**When** the system receives a request to calculate a benefits payment

**Then** the benefits payment amount is correctly calculated and displayed to the customer service agent

**Scenario 2:** Calculate benefits payment - Closed

**Given** that a customer has a benefits plan on file associated with their account

**And** the benefits plan is closed/not active

**When** the system receives a request to calculate a benefits payment

**Then** the system will return a message stating that the customer does not have an active benefits plan

**And** the system will return a benefit of \$0.00

**Scenario 3:** . . .

## Common Problems to Avoid

- **Not having a conversation**
  - A rich conversation with the team is needed to ensure a common understanding. Use whiteboards or other medium to communicate and ideate on the story.
- **Using User Stories for everything - even tasks**
  - Don't confuse stories (the 'Who', 'What', and 'Why') with tasks (the 'How')
  - Stating the 'how' in a story leads to limiting the solution. The team may come up with a better solution if they are aligned to the goal and have autonomy to find a good solution.
- **Undefined or wrong user (As a "user", As a "Developer", As a 'Product Owner', etc.)**
  - Example: "As a developer, I want to replaced the application widget, so that I have maintained the app widget." The Developer role is sometimes used to address issues of technical debt or spikes.
  - Identifying the correct user leads to a solution that better meets the needs of that user
- **No Acceptance Criteria**
  - Can cause the wrong definition of development tasks or the wrong estimation.
  - Story can fail tests or test cases will cover different criteria due to lack of understanding.
  - A good way of discovering Acceptance Criteria is asking questions such as 'What if ... ?', 'Where ...?', 'When ...?', 'How will we know we're done?'.