

# TURN USER STORIES INTO COHERENT TESTABLE REQUIREMENTS

By Mal Stiefel

[malstiefel@mindspring.com](mailto:malstiefel@mindspring.com)

978-407-5064

SQGNE: May 11, 2016

## Objectives

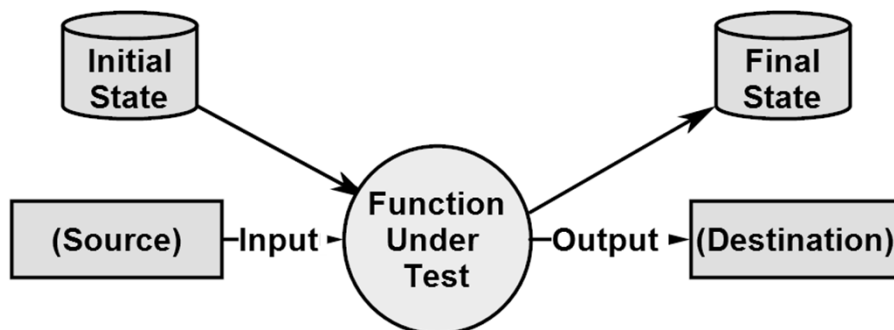
- How to transform user stories into use cases
- How to use state information to establish relationships among use cases and ensure completeness of a set of functional requirements
- How to leverage an inherently testable model of functional requirements

## Properties of Well-Written Requirements

- Clear (unambiguous)
- Complete
  - *Eventually* cover all functions and performance of system to be built
- Consistent
- Correct
- **Testable**
- Traceable

## Testable Functional Requirement

- Inputs and sources
- Outputs and destinations
- Initial and final states of the element being tested

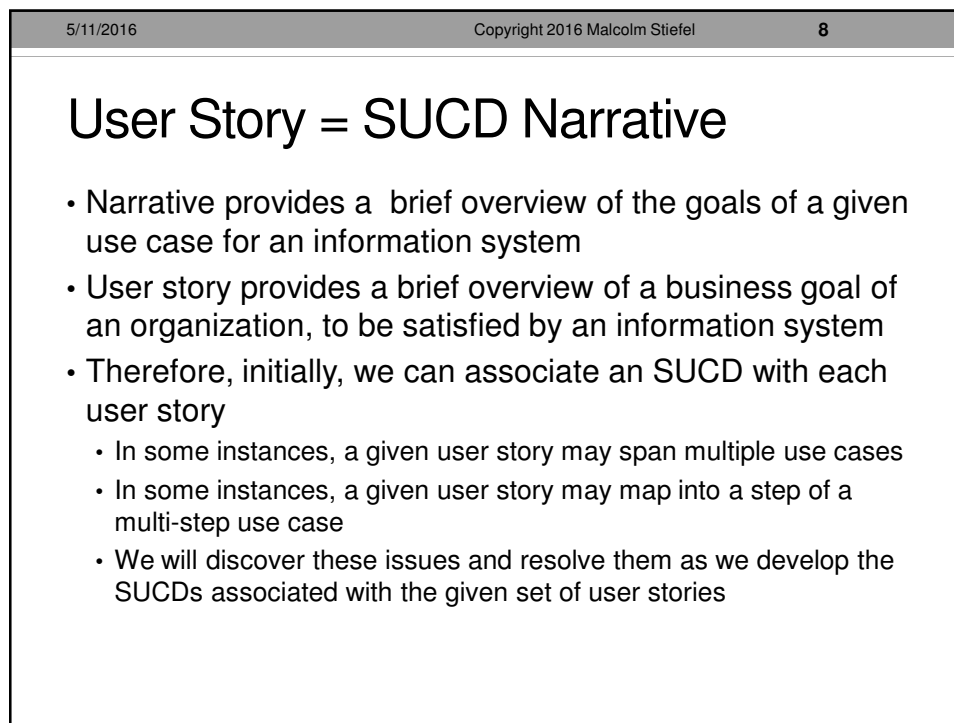
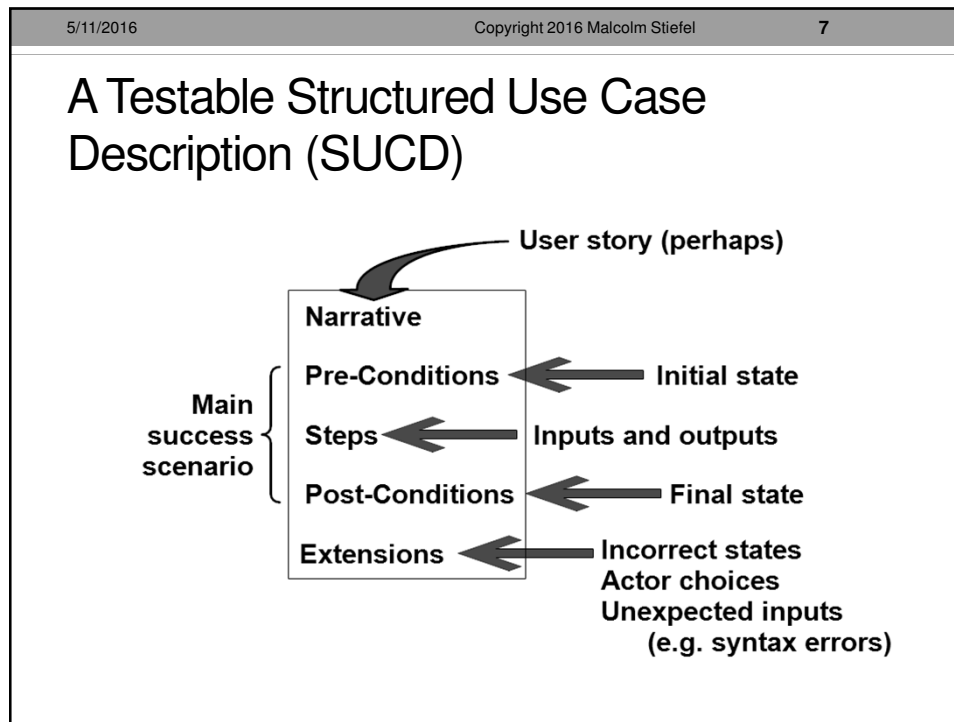


## User Story

- As an <actor name> I want <some business functionality> so that <business objective>
- Example: The Auction Program (TAP)
  - As a **Bidder** I want to offer a **bid** for a specific **item** offered for auction so that I can buy the item

## Testability Analysis

Testable Ftn Property	Value
Input information	Bid amount; item ID (implied)
Input source	Bidder
Initial state	Item ID exists (implied) Current high bid amount TBD Minimum Bid increment TBD
Output information (1)	Bid amount (implied) Item ID (implied)
Output destination (1)	Auction data store (implied)
Output information (2)	Indication that bid has been accepted TBD
Output destination (2)	Bidder TBD
Final state	Unchanged item ID (implied) Updated high bid amount (implied)



## User Story = SUCD Narrative (continued)

- It is likely that the set of use cases defined by a given set of user stories will be incomplete
  - That is, the information system may require additional functionality, not addressed in any user story, to create the set of desired states
- Therefore, we need some additional analysis mechanism to discover these additional SUCDs
  - And we will write additional user stories as their narratives

## Sample SUCD: Narrative

- A logged-in bidder searches for items, whose auctions are underway, to bid on, selects an item, and submits a valid bid, which TAP stores for the selected item.
- As an alternative, we could copy the user story into the narrative.
  - As a Bidder I want to offer a **bid** for a specific **item** offered for auction so that I can buy the item
  - User story in this example has less detail than the narrative above
    - Narrative above gives the reader a better idea of use case scope

## Sample SUCD: Pre-Conditions

- Bidder is logged on
- At least one item has an auction underway
- Each item offered for auction has a description; a minimum bid price; a current high bid value greater than or equal to the minimum bid price; and a minimum bid increment.

## Sample SUCD: Steps

Input from Actor	System Response
1. Bidder requests capability to search for and bid on items being auctioned.	i. TAP determines that Bidder is logged on. ii. TAP provides capability for Bidder to search for and bid on items being auction, showing for each item: description, current minimum acceptable bid price (see * below), remaining hours/minutes of auction. iii. TAP determines that an auction is currently underway for at least one item.  *Minimum acceptable bid price = larger of: a. Minimum bid price if no previous bids have been entered for this item; or b. Previous bid plus minimum bid increment, if one or more previous bids have been entered for this item.

## Sample SUCD: Steps (continued)

Input from Actor	System Response
2. Bidder searches for item to bid on	i. TAP keeps track of current item being examined by Bidder.
3. Bidder designates item to bid on, and enters bid amount.	i. TAP determines that entered bid amount is greater than or equal to minimum acceptable bid price ii. TAP stores entered bid amount as current high bid amount for designated item. iii. TAP provides transaction details to TAP audit log iv. TAP provides acknowledgment to Bidder that bid has been accepted v. TAP asks Bidder, "Do you want to search for another item?"
4. Bidder declines to search for more items.	i. Use case terminates.

## Sample SUCD: Post Conditions

- Current high bid amount for designated item is equal to entered bid amount
- Bidder is owner of current high bid amount for designated item
- Audit log is updated with transaction details

## Sample SUCD: Partial Extensions List

- 1a: Bidder is not logged on
- 1b. No items currently being auctioned
- 3a. Syntax error in entered bid
- 3b. Entered bid amount is less than minimum acceptable bid price
- 3c. Auction ends before entry is made.
- 4a. Bidder indicates intention to search for another item

## What Makes an SUCD Testable?

- **Each step** satisfies the testability properties
- Pre-conditions of main success scenario constitute initial state of first step
- Final state of first step is initial state of second step
  - And so on
- Post conditions are collection of all differences between initial state of first step an final state of last step
  - If a use case only reads data but does not change it, post-conditions are null



User Story	SUCD
Not testable	Testable
Very easy (low cost, fast) to create and maintain	Harder (higher cost, slower) to create and maintain
May rely on continued availability of originator to answer questions, explain intent	Does not rely on continued availability of originator
Requires additional effort to produce acceptance test cases	Already written in the form of acceptance test cases
<your notes here>	

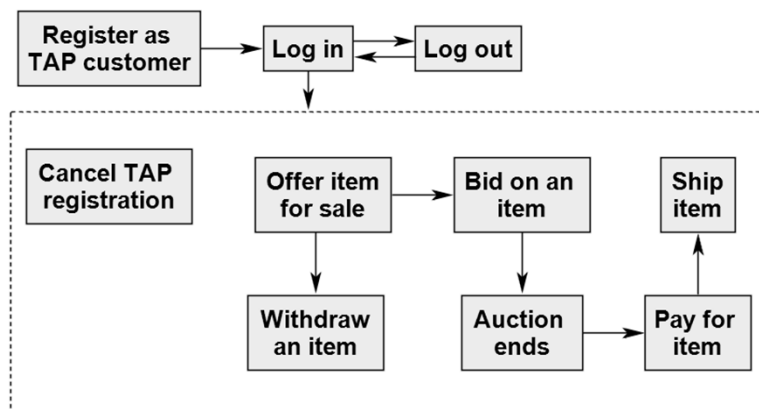
5/11/2016	Copyright 2016 Malcolm Stiefel	18
<h2>Use Case Dependencies</h2> <ul style="list-style-type: none"> <li>Initial state of a use case may be: <ul style="list-style-type: none"> <li>Combination of final states of previous use cases</li> <li>Null</li> </ul> </li> <li>In our example: <ul style="list-style-type: none"> <li>An item (offered for sale) exists as part of our use case initial state</li> <li>Existence of the item results from execution of an earlier use case: offer item for sale.</li> <li>Therefore, successful execution of “Bid on an item” use case depends on successful prior execution of “offer item for sale” use case</li> </ul> </li> </ul> <div style="text-align: center;"> <pre> graph LR     A[Offer item for sale] --&gt; B[Bid on an item] </pre> </div>		

## What Happens Next?

- The question (what happens next?) probably needs to be addressed by another user story
  - It involves a different actor: the Successful Bidder, or the Winner
  - It probably involves another actor as well: a TAP administrator
  - It occurs after the auction ends, not during the auction

## Use Case Dependencies (continued)

- We can use similar reasoning to identify additional use cases:



## Summary

- Transform untestable user stories into inherently testable SUCDs
- Each SUCD can be instantiated as an acceptance test scenario
- Use dependencies among SUCDs to identify all use cases in an application