

Agile 2008

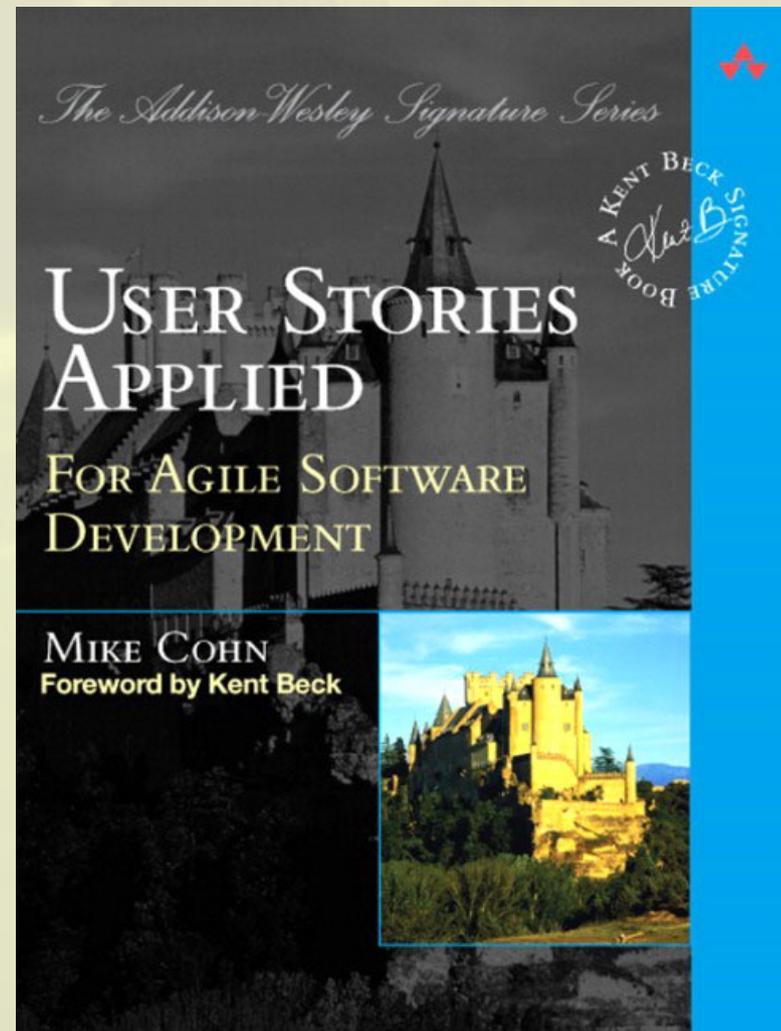
User Stories and Use Cases?
Sure, it even makes sense.

What do I mean by a User Story?

- A user story is similar to a feature and captures the system from the user's perspective
- User stories can be used to uniformly capture both functional and non-functional requirements
- User stories represent a sized unit of value delivered to the user
- User stories are sized relatively using a finite sized, fixed, non-linear scale
 - 1, 3, 5, 8, 13, 20, 40, 100
- Dependencies will exist between some user stories, however stories should be sized independently
- User stories should have an end user that consumes an interaction with the system
- Stakeholders have a good chance of understanding user stories

Check out “User Stories Applied” by Mike Cohn

- This text provides detailed look at user stories
 - A better place to start than an XP text



Identify a Finite, but evolving, set of End Users

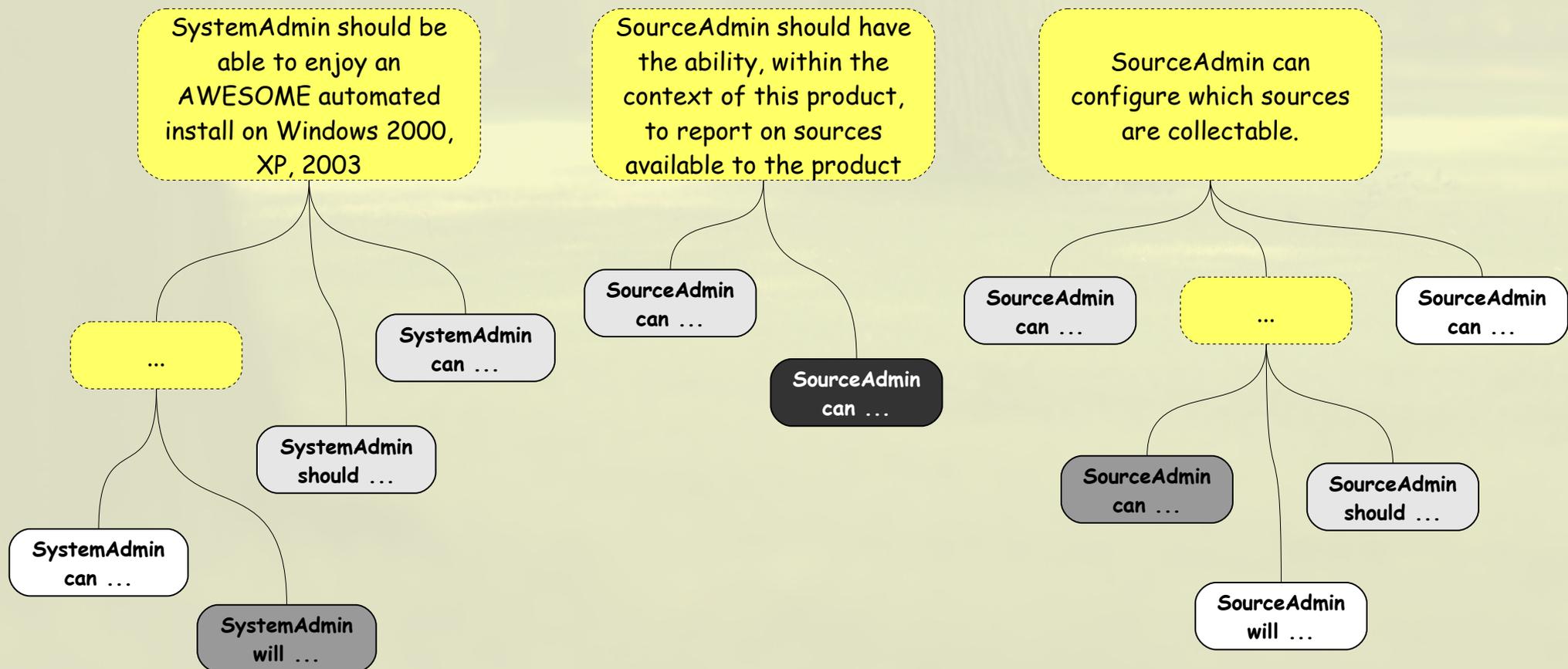
- Careful identification of User classes is an important skill requiring stakeholder input
- A shared understanding of the who the Users of a system are is central to communicating effectively the User Stories with the Customer.
- Some typical users might be
 - SystemAdmin
 - SecurityAdmin
 - EndUser
 - ProductSupport
- As understanding of a product improves User classes can be split or refined as appropriate
- Some End Users will have very specific business domain roles

User Story Detail and Splitting Large User Stories

- Initially user stories will be a simple statement of what the user wants to achieve with a relative size
 - When stakeholders wish to consider a user story for development a completion statement must also be provided
- Over time any dialogue associated with the user story will also be captured
 - This may include relevant use cases and test cases
- Typically user stories above a certain relative size cannot be implemented within a single iteration
 - In that case the user story must be split into two or more user stories
 - This is sometimes difficult but decreases risk
- As user stories are split the resolution of the project is improved
 - Retaining the history of splits allows visualising a capability view

Splitting User Stories helps build a Hierarchical View of the Project

- As user stories are split the tree gets broader and sometimes deeper
- The high level user stories can be grouped into capabilities

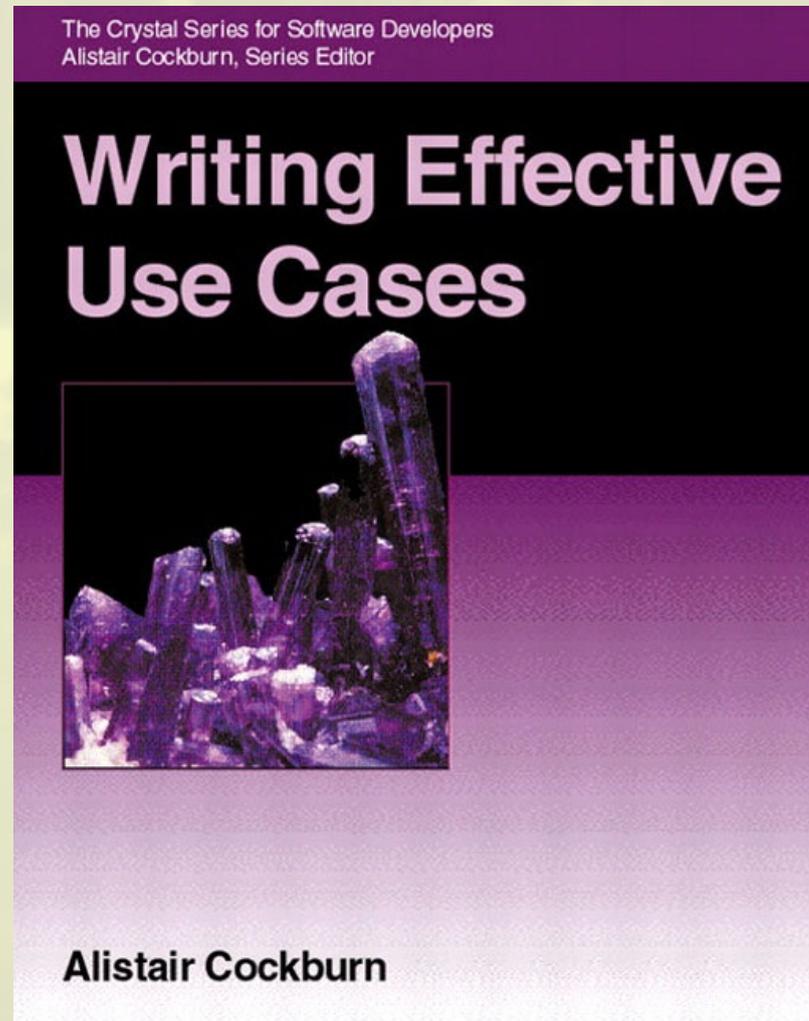


What do I mean by a Use Case?

- A Use Case is a textual artifact that specifically examines one or more scenarios from a User's perspective in the User's attempt to accomplish some specified goal.
- A Use Case
 - Exists in a certain Design Scope
 - Is enacted by the identified User
 - The User is trying to achieve a Goal
 - To achieve the Goal the User must complete a Main Success Scenario
 - Certain Pre-Conditions may be required before starting
 - Completion of the scenario may have Guarantees
 - There may be extension scenarios for deviations

Check out “Writing Effective Use Cases” by Alistair Cockburn

- Most of the information here has been adapted from information in Alistair's book
 - If you haven't read it find time to do so



A Use Case Template

- The template shown is a compromise between brevity and value
- The Use Case Summary Title (USE CASE NAME) captures the essence of the Use Case
- The Main Success Scenario will have at least 2 steps but not too many (say 3 to 10)
- Extensions refer to specific steps in the Main Success Scenario
 - More than four levels deep here implies another Use Case

USE CASE ##	Design Scope Icon	USE CASE NAME	Goal Level Icon
Goal in Context	< a longer statement of the goal, if needed, its normal occurrence conditions >		
Scope	< design scope, what system is being considered black-box under design >		
Level	< user goal level >		
Primary Actor	< a role name for the primary actor, or description >		
Stakeholders & Interests	< list of stakeholders and key interests in the use case >		
Preconditions	< what we expect is already the state of the world >		
Minimal Guarantee	< how the interests are protected under all exits >		
Success Guarantees	< the state of the world if goal succeeds >		
Trigger	< what starts the use case, may be time event >		
Main Success Scenario			
1	First step		
2	Second step		
3	Third step		
Extensions			
*a Condition on whole			
	*a1. Action 1 with link?		
2a. Condition on step			
	2a1. Action on step		
	2a1a Condition on action		
	Last action depth - deeper than this needs it's own use case		
2b. Second condition on same step			
	2b1. Action		
	2b2. Second action		
Technology & Data Variations			
1. The specified item can be:			
	* item 1		

Choose a Use Case Template and use it consistently

- Scope and Level are important so are also represented with icons for quick browsing
- The Primary Actor will be one of our previously identified users, bridging the gap between User Stories and Use Cases
- Technology and Data Variations are used to detail 'how' something is different
 - Extensions capture 'what' is different from the Main Success Scenario

USE CASE XXXX	Design Scope	Use Case Summary Title	Goal Level
Goal in Context			
Scope			
Level			
Primary Actor			
Stakeholders & Interests			
Preconditions			
Minimal Guarantee			
Success Guarantees			
Trigger			
Main Success Scenario			
1			
n			
Extensions			
1a.			
1a1.			
Na.			
Technology & Data Variations			
1			
n			
Comments			

Use Cases have a Design Scope

- Identifying a design scope helps remove ambiguity
 - Business Use Cases for the Organisation/Enterprise
 - We are discussing the behaviour of the entire organization or enterprise in delivering the goal of the primary actor
 - Black box if we are treating the organisation as a black box
 - White box if we are discussing departments
 - System
 - We are discussing the behaviour of the system we are building
 - Black box if we treat the system as a black box
 - White box if we reveal details of the internals
 - Sub-system
 - We are discussing a sub-system or component of the system we are building

Design Scope	
Icon	Meaning
	Organization (black-box)
	Organization (white-box)
	System (black-box)
	System (white-box)
	Component

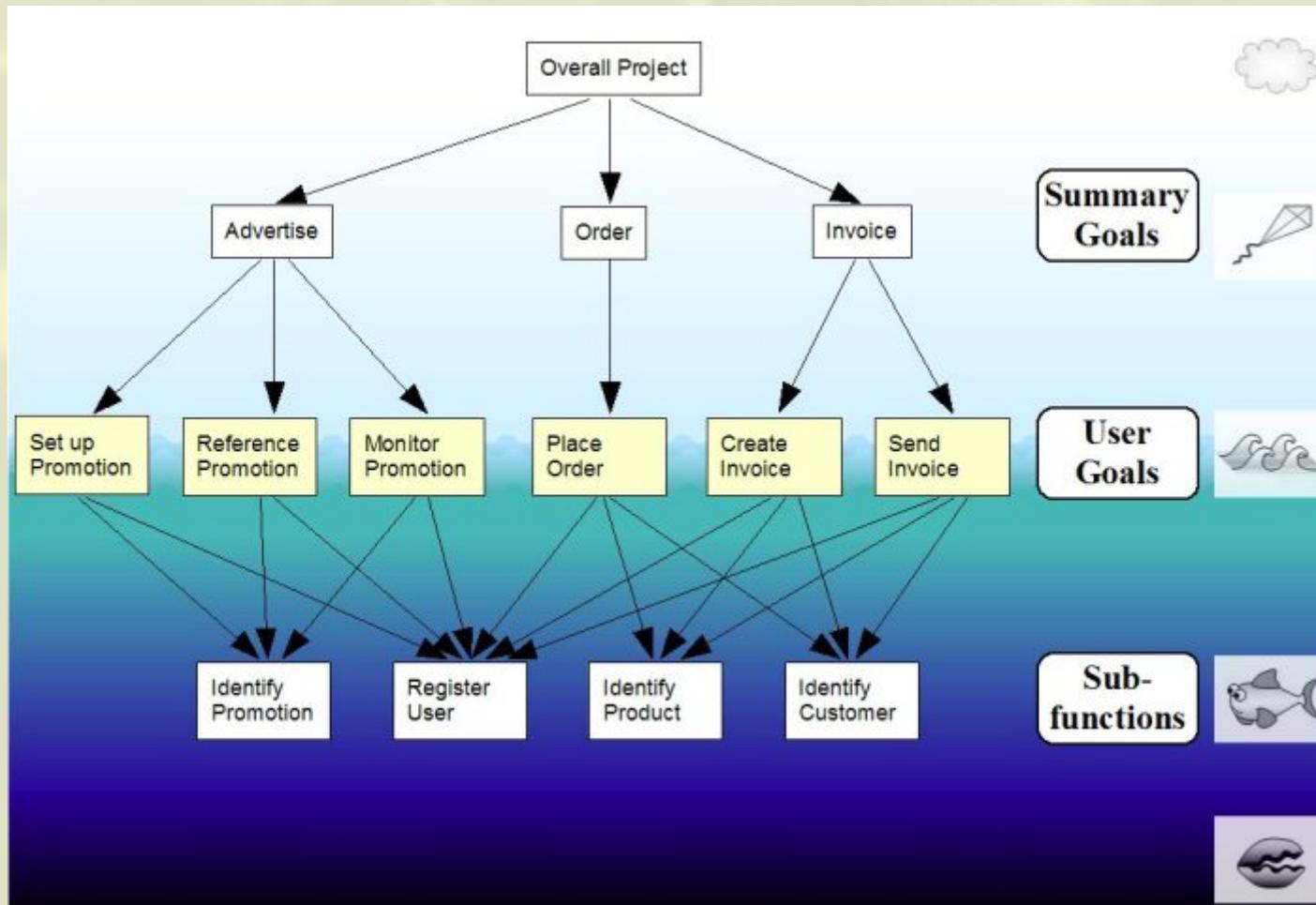
Use Cases have a User Goal Level

- Every Use Case has a User Goal Level
 - We identify 5 goal levels for Use Cases
- Higher level goals can be unfolded or split into one or more lower level goals
 - Higher level goals provide context for lower level ones
- User Level Goals are the most important and most energy should be spent trying to detect these use cases
 - Subfunction Goals should only be added if they are needed to remove ambiguity
- The Goal Level icons use a Sky-SeaLevel-Sea metaphor to illustrate that most use cases will appear at sea level. There is only one sea level.

Goal Level	
Icon	Meaning
	Very high summary
	Summary
	User Goal
	Subfunction
	Too Low

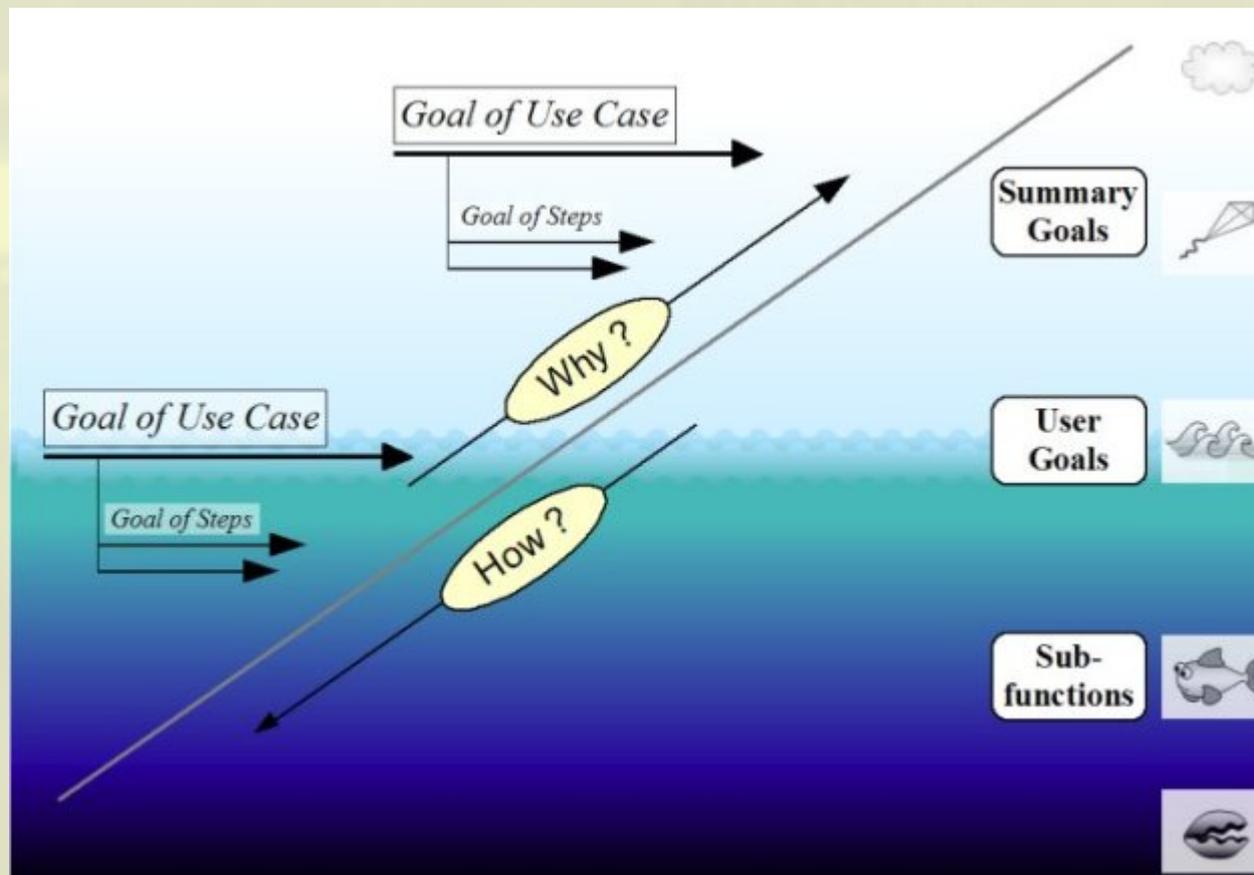
Use Case Goal Levels: Pictorially

- Use Case goals levels reflect the different usage levels in a system



Use Case Goal Levels: Emphasis changes as level changes

- Higher level goals address Why, lower level goals address How
 - To move down a level ask How?
 - To move up a level ask Why?



Use Cases are UI Agnostic

- Emphasis is on WHAT is achieved, not the mechanics of HOW it is achieved
- User Documentation describes the 'how'
- Test Cases for User Interactions can be written in two phases
 - First phase ensures the Use Case path is clear
 - Second phase adds details based on User Doc
- Acceptance tests can validate both the UI and the User Documentation independently
- Use Cases are therefore very useful when used to capture complex User Interactions

Use Case Summary and Sample

- The Use Case clearly and succinctly provides an interaction capability promise
 - Preconditions and Guarantees are important
 - The Main Success Scenario is not overly long
 - There is no UI information
 - There shouldn't be. Such detail depends on the implementation
 - The Extensions can be implemented and tested separately

USE CASE 201  **The user needs to be able to configure initial host(s)** 

 Edit

Goal in Context	The user needs to be able to configure initial host(s)
Scope	NMS
Level	User-Goal
Primary Actor	System Administrator
Stakeholders & Interests	System Administrator, NMS Administrator, NMS
Preconditions	Install must be in progress
Minimal Guarantee	System Administrator gets a message
Success Guarantees	Host(s) get configured
Trigger	System Administrator selects to configure the host(s) during the install

 Edit

Main Success Scenario

- 1 Sys Admin elects to configure an initial host
- 2 Sys admin enters [Host Information](#) (repeatable)
- 3 Sys admin submits [Host Information](#)(may return to step 3)
- 4 Success message sent to sys admin; install continues.

 Edit

Extensions

edit	3a Host already entered
edit	3a1. Message to admin, return to 3 or continue.
edit	3b Invalid Host Information
edit	3b1. Send message; Return to 3
edit	3c. Unable to add Host Information
edit	3c1. Message to Admin

 Edit

Comparing User Stories and Use Cases

■ User Stories

- They are relatively sized and exist at all levels of implementation
- They are a proportional unit of progress (value delivered to the customer) and are either DONE or NOT DONE
- Detail is deferred both at the milestone level and the iteration level
- User stories are lightweight to write initially

■ Use cases

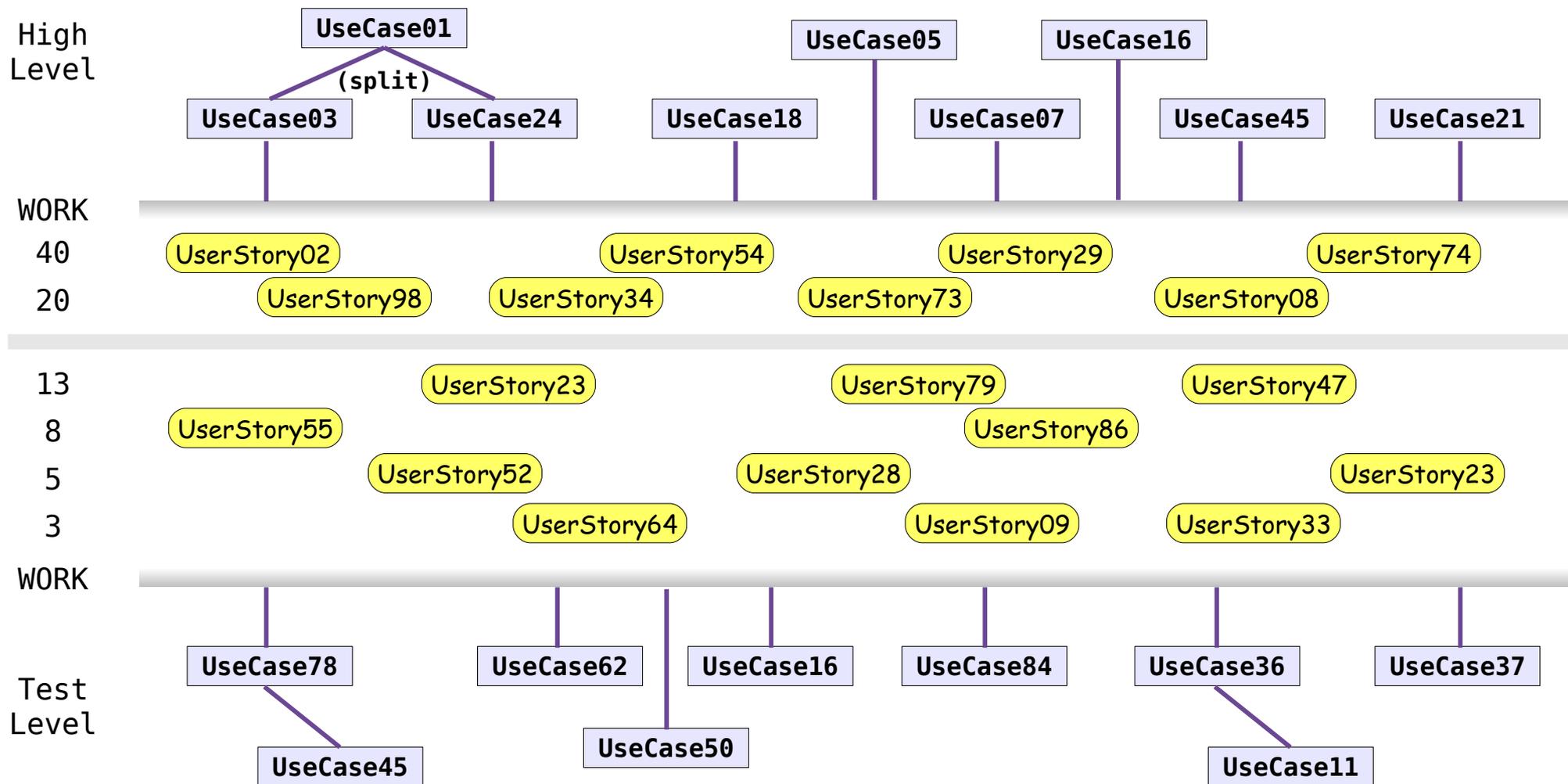
- They are not sized and are not used to measure progress
- They exist at all levels of the product and capture goal driven scenarios enacted by a user
- Their purpose is to clarify understanding of the system from the user's perspective
- Use cases are more heavyweight to write initially

Understanding the Relationship between User Stories and Use Cases

- Use cases can be seen as way to provide a multi-level representation of the project's capabilities
 - High level use cases might be used to help with strategic planning
 - More detailed use cases could be used to allow the development of user acceptance test plans
- How do Use Cases relate to User Stories?
 - To some degree we can think of User Stories as Use Case Briefs at a particular level of detail
 - The primary difference is that a use case is scenario based
 - Use cases that relate to one or more user stories can be thought of as the context in which the user stories are enacted
 - Conversely a use case could be an elaboration of a user story
 - However we should focus not on what Use Cases and User Stories say, but rather on how they are used

Visualising User Stories and Use Cases

- Arranging 2-dimensionally user stories and use cases we might end up with something like this:



Levels of User Stories and Use Cases

- Conceptually three basic levels at which we can describe the product
- Use Case Briefs
 - We have high level use cases that may potentially encompass one or more user stories
 - To understand what will be delivered consult the user stories
- User Stories
 - Relatively sized user stories: Potential work items that can be scheduled for implementation
 - Smaller user stories that can be targeted for work in an iteration
- Use Cases
 - Lower level use cases specify a specific success scenario that may be relevant to one or more user stories

User Stories and Use Cases: Complimentary Tools that Work

- User Stories can be thought of as a release planning tool
 - They help understand progress
 - They tell us what value we have delivered to stakeholders
- Use Cases can be thought of as an iteration tool
 - They can provide a broader context for user stories
 - They can provide more detail to help develop test cases
- User stories may be related to one or more use cases
- Use cases may be related to one or more user stories
- Both tools provide value to the project without introducing unnecessary redundancy
- User stories and use cases provide complementary information helping stakeholders and team members gain a more complete view of the project

Questions?