



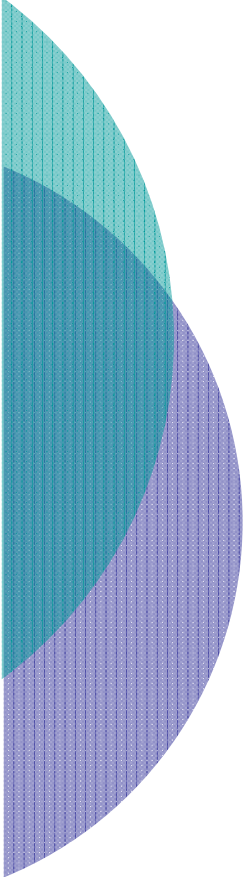
Scenario-Based Testing added to an Existing Development Program

GSAW

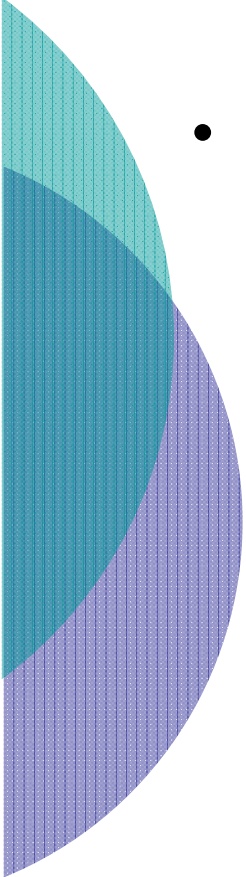
April 1-3, 2008

A. Shaw

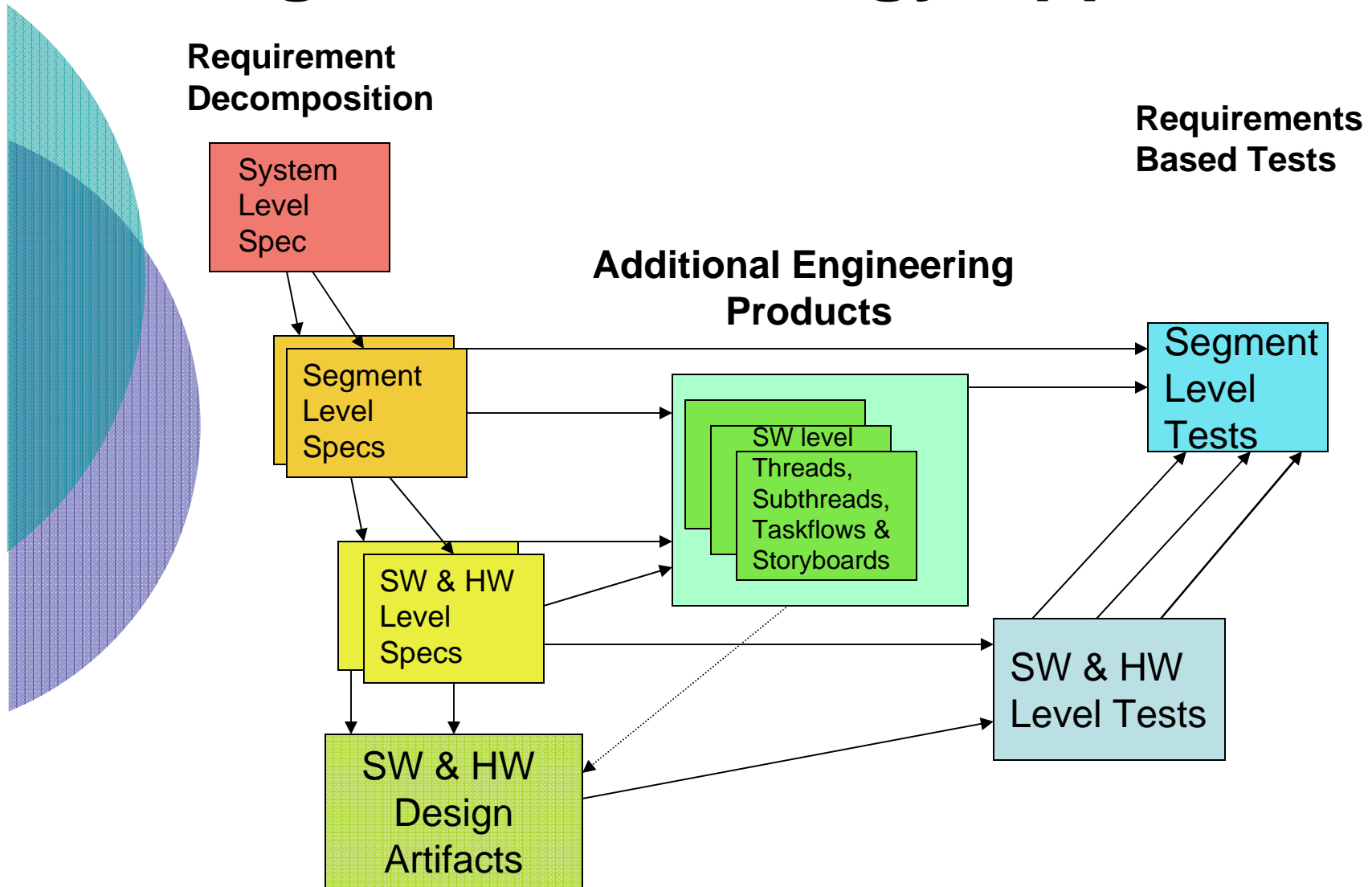
Presentation Overview

- 
- **Long-term Project with
older development methodology**
 - **New development Methodologies introduced
(example included)**
 - **Benefits of revised Methodology**

Original Methodology

- 
- **Existing Development Program**
 - **Long running development project** using several stages (iterations) of development and delivery to customer over more than a decade
 - **Requirements decomposition methodology:**
 - Used System level Spec to Segment level Spec to Software Item (SI) and Hardware Item (HI) level Specs
 - Developed Threads and sub-threads describing how the system operates for specific tasks
 - Developed Task Flows and storyboards for operator interaction
 - **Issues with approach**
 - SI oriented, system view limited
 - Focus of engineering is on end result, not on interim releases
 - Development staging an add-on, not integral to engineering

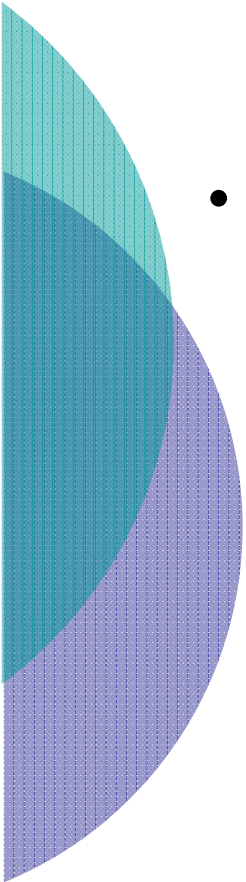
Original Methodology Approach

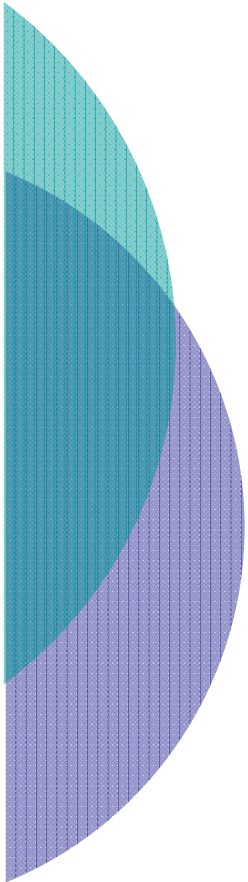


4/21/2008

Unclassified

Original Methodology Results

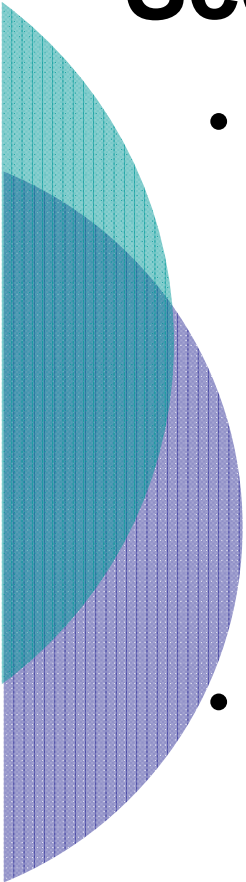
- 
- **Existing Development Program results:**
 - **Threads focused on domains →**
Integration&Verification (I&V) thread-based testing
focused short set of steps
 - **Domain-focused testing →**
System not ready to execute operational activities
 - **Minimal system testing performed before the day-in-the-life**
validation test preparation activities →
System stability issues and schedule impacts



Some Background Information

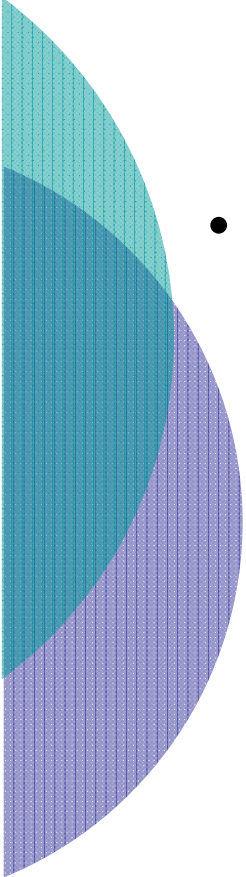
Scenario definitions

Scenario-based Testing Definitions

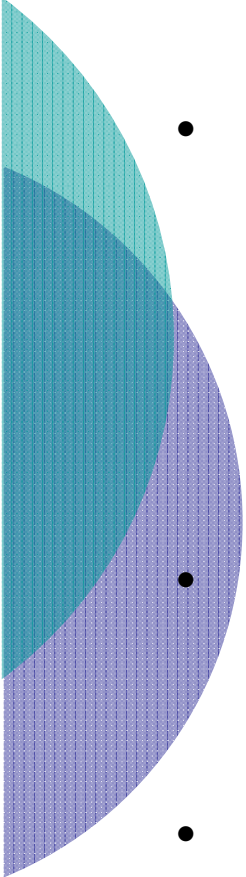
- 
- **Scenario-based testing (SBT) is:**
 - Verification of system functionality by executing test scenarios or test cases that are associated with requirements and with system operational flows
 - An analytic means to achieve high confidence that the test program “closes” on a suite of tests, demonstrations, analyses and inspections that will fully verify the operational system capabilities
 - **Scenarios are (*):**
 - Based on **a story** about how the program is used
 - Based on what a **stakeholder cares** about
 - **Credible**, i.e. something that probably will happen
 - Defined so that **results** are **easy to evaluate**

(*) From paper by Cem Kaner, **An Introduction to Scenario Testing.**
June 2003, Unclassified

Scenario-based Testing

- 
- **Purpose of developing Scenarios:**
 - **Validate the requirements baseline**
 - **Engineering stage**
 - **Validate the design artifacts**
 - **Design stage**
 - **Verify requirements as artifacts associated with the architecture (using scenarios)**
 - **Testing stage**

Scenario Variations

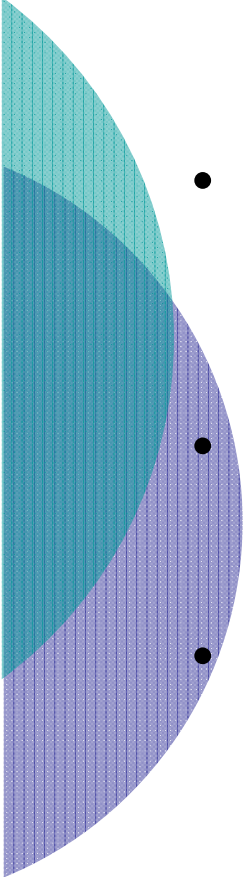
- 
- **Some scenarios may have variations, which represent alternate flows through the scenario, usually associated with error conditions.**
 - **Candidate variations are identified by examining the threads associated with the requirements and**
 - **by considering operational experience.**
 - **Variations are accepted if they are needed to test some of the scenario's requirements or if they exercise the software in a unique way.**
 - **Variations are not created for multiple simultaneous failures unless there are requirements to deal with such failures.**
 - **Scenario Variations are usually considered separate test cases of the SBT**

Engineering Methods Updated to include Scenario Based Tests

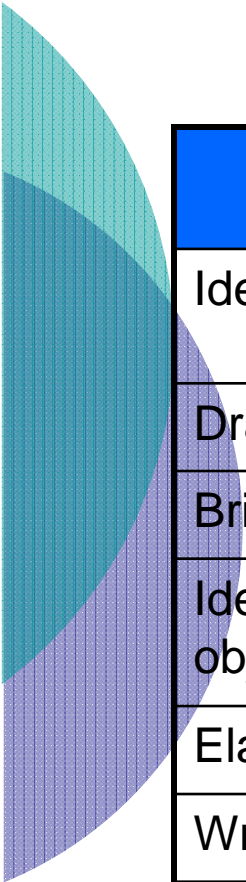
- **Scenario definition**

- **Define candidate scenarios using requirements baseline and capabilities for each release**
 - **Make up a list of candidate scenarios based on the delivery plan**
 - **Make up a list of candidate segment requirements for each scenario**
 - **Check for overlap between scenarios**
 - **Choose the best scenario for each requirement, redefining scenarios as needed to produce a set of scenarios that**
 - **cover all requirements,**
 - **are of manageable size, and**
 - **describe the expected use of the system**

Engineering Methods Updated for Scenario Based Tests

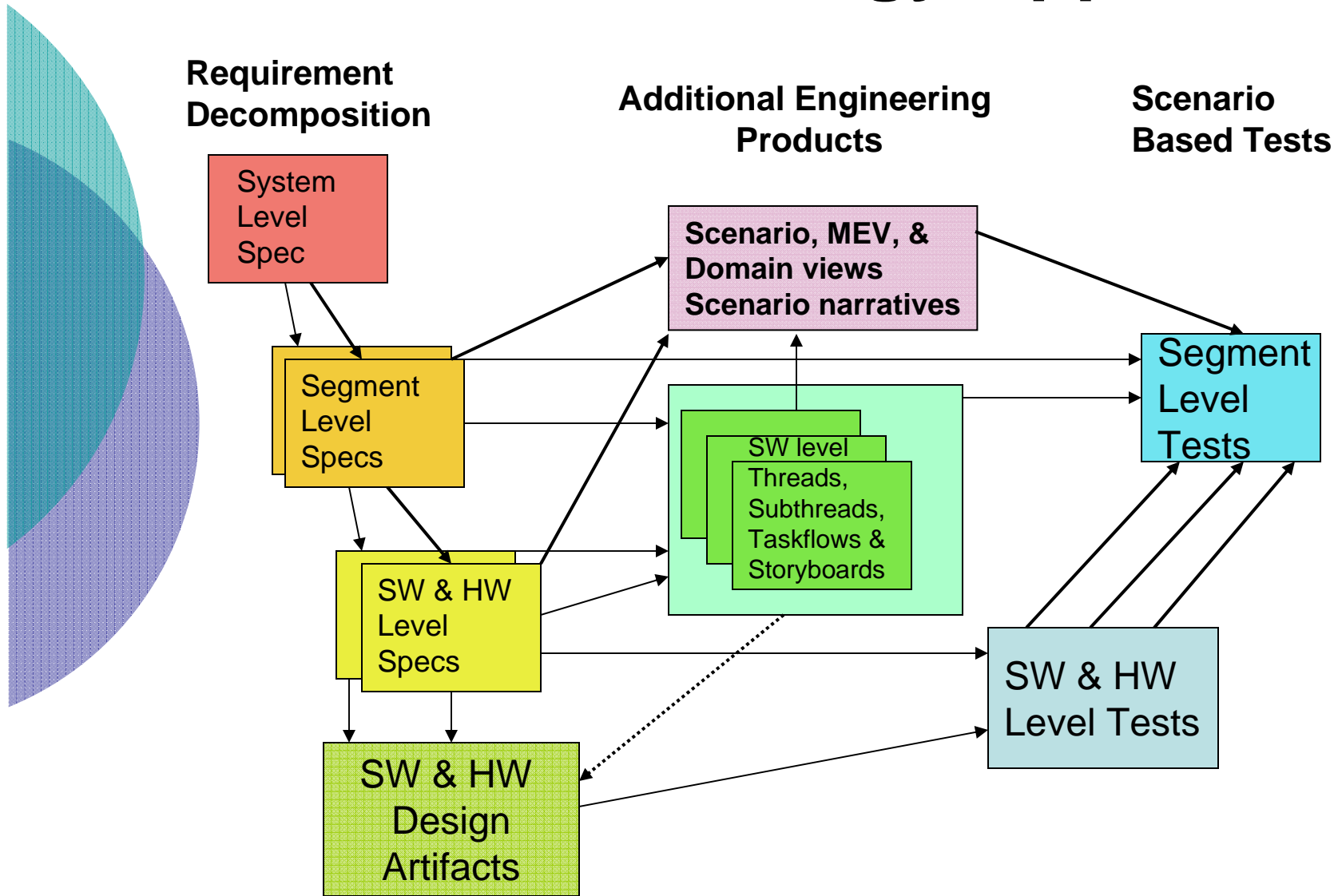
- 
- **Mission executable view**
 - the interactions among the architectural functions in the scenario
 - extracted from the segment functional architecture
 - **Domain view**
 - the interactions among the hardware and software domains in the scenario extracted from the domain architecture
 - **Scenario narrative**
 - the table of steps that describe the scenario operationally, that can be incorporated into the test plan and expanded into a test case
 - by requirements,
 - by interfaces and
 - by threads

Scenario based Test Plan Development Steps



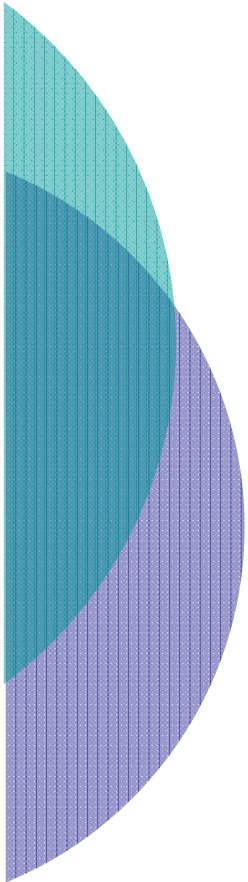
| Step | Responsibility |
|--------------------------------|---|
| Identify scenarios | Chief Systems Engineer, Verification Lead, Chiefs |
| Draw scenario diagram | Systems Engineer |
| Brief scenario diagram | Systems Engineer |
| Identify additional objectives | Systems Engineer, Verification tester, Software Engineer |
| Elaborate objectives | Systems Engineer, Verification tester |
| Write narrative scenario | Systems Engineer |
| Write test plan | Verification tester |
| Review test plan | Systems Engineering, Verification tester, Software Engineer |

Revised Methodology Approach



4/21/2008

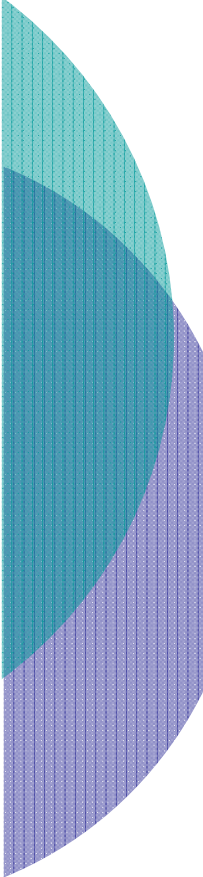
Unclassified



Scenario Based Test Example

Bachelor Cook

Bachelor Cook Requirements

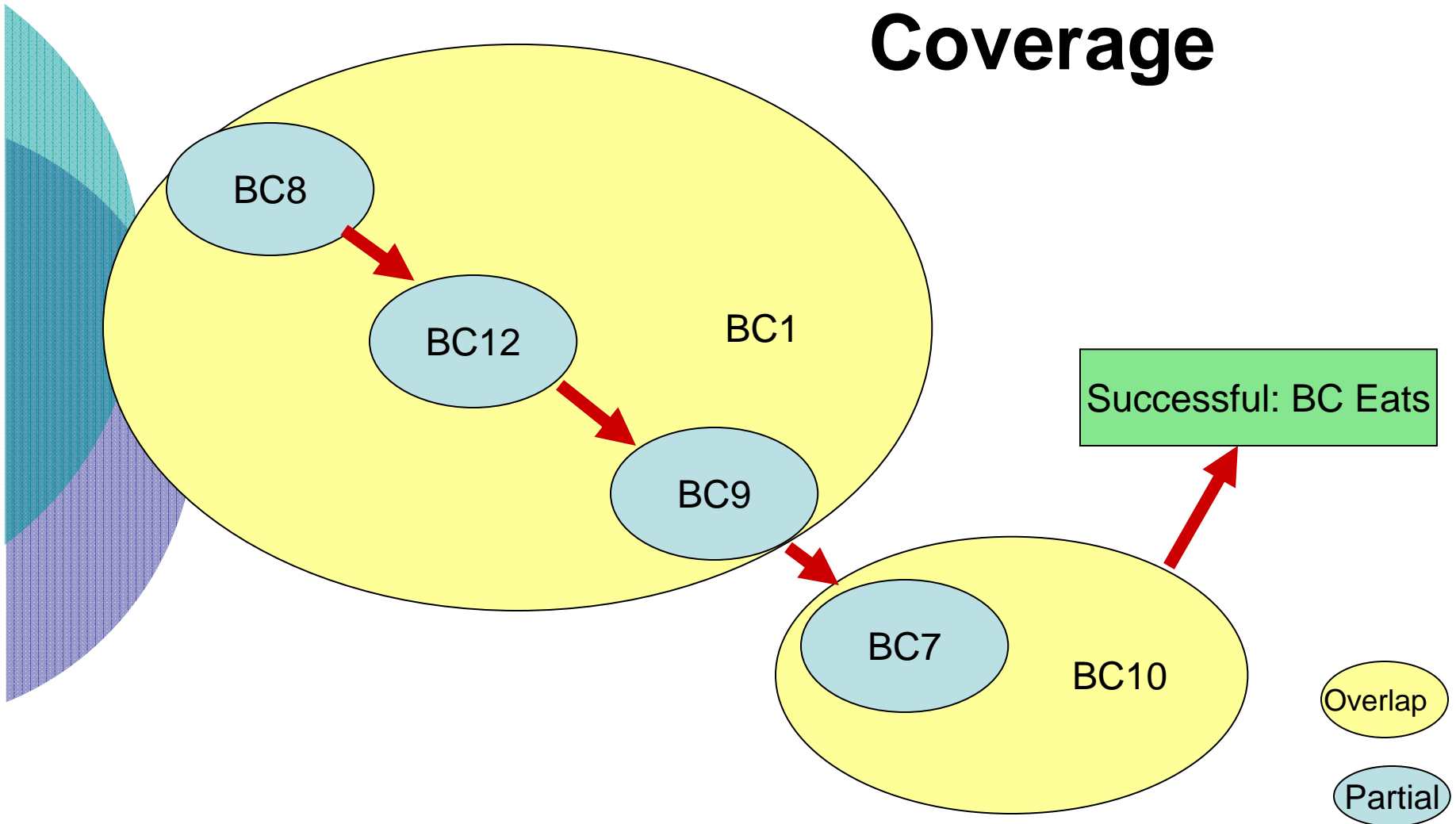


| Reqt ID | Requirement Text |
|---------|---|
| BC1 | The Bachelor Cook shall be able to prepare a PBJ. |
| BC2 | The Bachelor Cook shall be able to prepare canned soup. |
| BC3 | The Bachelor Cook shall be able to prepare spaghetti. |
| BC4 | The Bachelor Cook shall be able to prepare hamburger. |
| BC5 | The Bachelor Cook shall be able to use a cook top. |
| BC6 | The Bachelor Cook shall be able to use a sharp knife |
| BC7 | The Bachelor Cook shall be able to clean implements. |
| BC8 | The Bachelor Cook shall be able to purchase ingredients. |
| BC9 | The Bachelor Cook shall be able to use a butter knife, spoon, fork and spatula. |
| BC10 | The Bachelor Cook shall be able to clean the kitchen. |
| BC11 | The Bachelor Cook shall be able to assess freshness. |
| BC12 | The Bachelor Cook shall be able to assemble required ingredients. |
| BC13 | The Bachelor Cook shall be able to prepare variations of core recipes. |

Bachelor Cook Scenarios

| Scenario | Test Case | Description |
|-------------|-----------|---|
| SC1 | | Prepare PBJ – Req'ts = BC1, 7, 8, 9, 10,12, 13 |
| | SC1.1 | PBJ, Successful, smooth peanut butter, grape jelly |
| | SC1.2 | PBJ, Failure, Missing ingredients - bread |
| | SC1.3 | PB, Successful, no jelly |
| SC 2 | | Prepare Canned Soup - Req'ts = BC2, 5, 7, 8, 9, 10,11, 12, 13 |
| | SC2.1 | Soup, Successful, Noodle |
| | SC2.2 | Soup, Failure, Burnt |
| | SC2.3 | Soup, Successful, Tomato with Milk |
| SC 3 | | Spaghetti Dinner - Req'ts = BC3, 5, 7, 8, 9, 10,12, 13 |
| | SC3.1 | Spaghetti, Successful, Canned Sauce |
| | SC3.2 | Spaghetti, Variation, Burger |
| SC 4 | | Prepare Hamburger - Req'ts = BC4, 5, 6, 7, 8, 9, 10, 11, 12, 13 |
| | SC4.1 | Hamburger, Successful, Qty=1, Lettuce, Tomato, Onion |
| | SC4.2 | Hamburger, Successful, Qty=8, Cheese, Ketchup, Pickles |
| | SC4.3 | Hamburger, Failure, rancid meat |

Scenario Test Case Requirement Coverage



Note: Requirement BC13 is partial in Test Cases 1.2 and 1.3

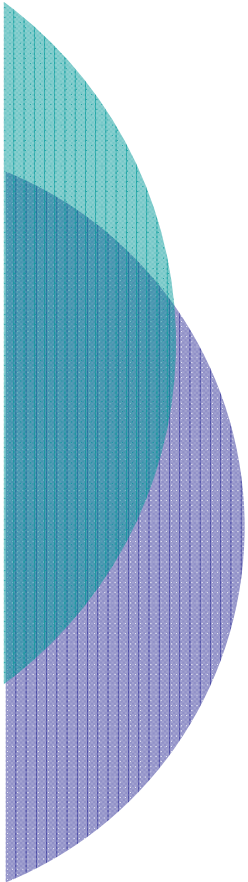
4/21/2008

Unclassified

Scenario Narrative –SC1.1

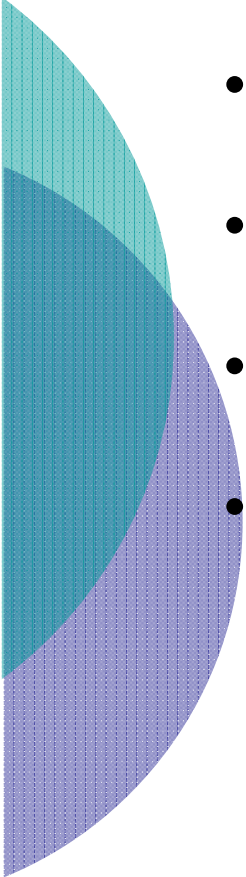
| STEP | Expected Results | Thread | Requirement | Interface |
|---|--|----------------------|-------------|--------------------|
| Purchase Peanut Butter, Jelly and Bread | All ingredients purchased | Purchase Ingredients | BC8 | External to store |
| Find Peanut Butter jar, Jelly jar and 2 slices of Bread | Cook has 1 jar of PB, 1 jar of jelly and 2 slices of bread | Assemble Ingredients | BC12 | Internal to fridge |
| Spread Peanut butter on one slice of bread | Peanut butter spread over 80% of bread slice | Use utensils | BC9 | N/A |
| Spread Jelly on other slice of bread | Jelly spread over 80% of other slice | Use Utensils | | N/A |
| Put two slices together with bare bread facing out | PBJ sandwich with PB and jelly on inside and bare bread on outside | | BC1 | N/A |
| Clean butter knife with soap and water and dry | Knife passes visual test for cleanliness. | Clean Implements | BC7 | Internal to sink |
| Wipe counter with wet dish cloth; rinse cloth in sink | Counter passes visual test for cleanliness | Clean kitchen | BC10 | |
| BC eats sandwich | Cook full and smiling | | | |

Unclassified



Conclusions

Benefits

- 
- **Adds to the confidence** of the program that the delivered System will meet the needs of the program.
 - **Uncovers problems in the baseline** that were not understood from the existing engineering – **validates design**
 - Uncovers **problems in the implementation** especially from the operator perspective (operability).
 - Extends beyond verification of individual capabilities to **validation** of system operability:
 - **A better segment test than RBT alone - tests the system as it operates**
 - **Forces early improvements in system maturity and stability**
 - **Focuses the development on the system not just individual capabilities**