Agile and Traceability

Bruce Powel Douglass, Ph.D.

Chief Evangelist, Global Technology Ambassador
IBM Rational
Bruce.Douglass@us.ibm.com
Twitter: @BruceDouglass
IBM: www-01.ibm.com/software/rational/leadership/thought/BruceDouglass.html

Go to IBM

Innovation for a smarter planet

Agenda

- On the importance of Traceability
- On the other hand …. Agile!
- Matching Traceability with the Agile Lifestyle

Traceability

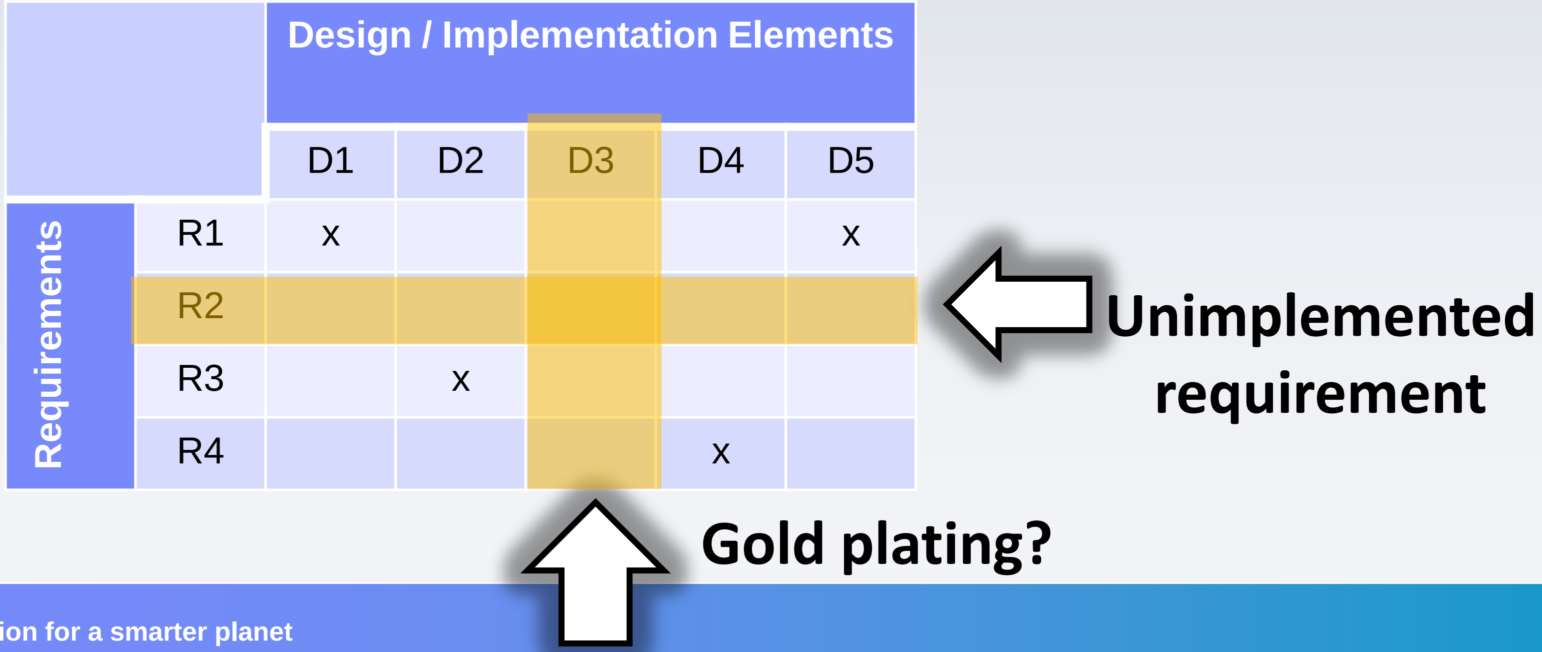
- ***Traceability*** *means that a navigable relation exists between all related project data, without regard to the data's location (within work products) or format*
- Forward Traceability
  - Data created earlier in the process can be navigated from to related (often derived) data produced later in the process
  - Examples:
    - Requirement → Design; Design → Implementation; Requirement → Test Case
- Backward Traceability
  - Data created later in the process can be navigated from to related (often source) data produced earlier in the process
  - Examples:
    - Design -> Requirement; Test Case → Requirement; Implementation → Design

Why Traceability?

- Impact Analysis
  - If I change this datum, what other project data is affected and must also be modified?
  - If I change this datum, what is the cost and effort required to make all relevant changes?
- Completeness Assessment
  - Have I (demonstrably) realized each datum?
  - E.g. Safety Objective, Requirement or design element
- Justification
  - Can I show why this data is present?
  - E.g. Is this design element present to meet a requirement?
- Consistency Evidence
  - Are the data within different aspects of the project data consistent?
  - E.g. Are the requirements consistent with the safety assessment? Does the implementation meet the design?
- Compliance Evidence
  - Does the data comply with internal and external standards
  - E.g. QA Records show process was followed (audit) or work product is well-formed (review)
- Required for Safety Critical – High Reliability – High Security systems
  - Safety standards require detailed traceability
  - E.g. DO-178 (Avionics), EN 50128 (Rail), IEC 62304 (Medical)

In Other Words, Traceability

- Helps ensure requirements are met
- Helps ensure requirements are verified
- Lowers project risk
- Creates an audit trail
- Ensure consistency among work products
- Helps manage consistency over the long term



		Design / Implementation Elements				
		D1	D2	D3	D4	D5
Requirements	R1	x				x
	R2					
	R3		x			
	R4				x	

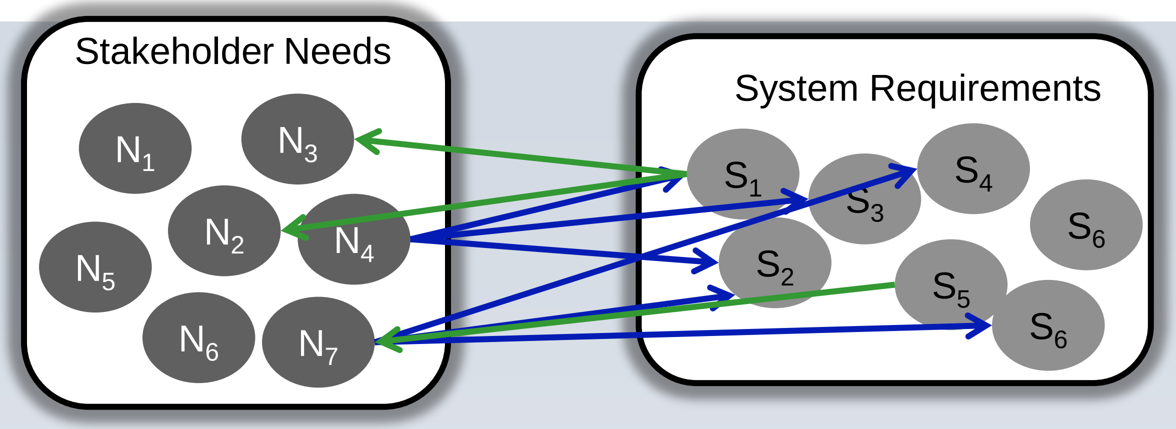
Unimplemented requirement

Gold plating?

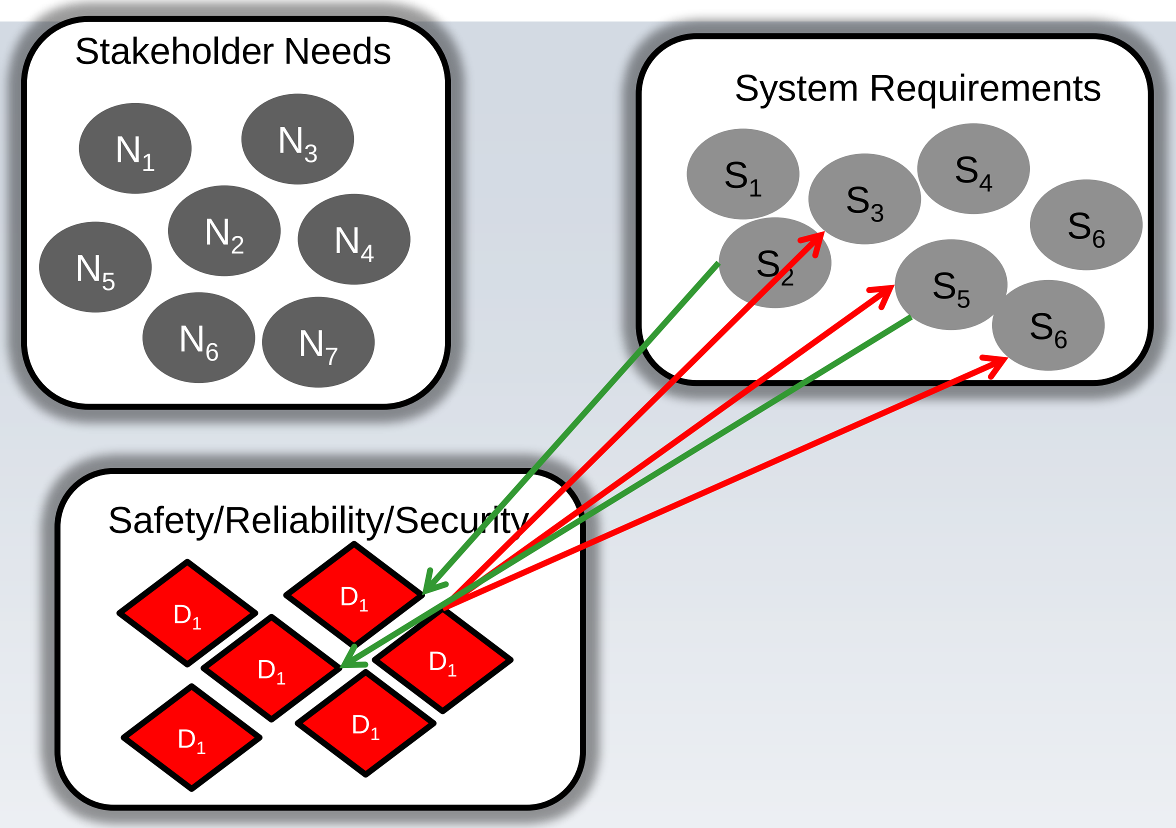
Why not Traceability

- Seen as low priority, low value work
- Arduous to create and maintain (esp without tools)
- Difficult to verify
- Work products may not be complete enough to support traceability
- May be difficult to demonstrate ROI

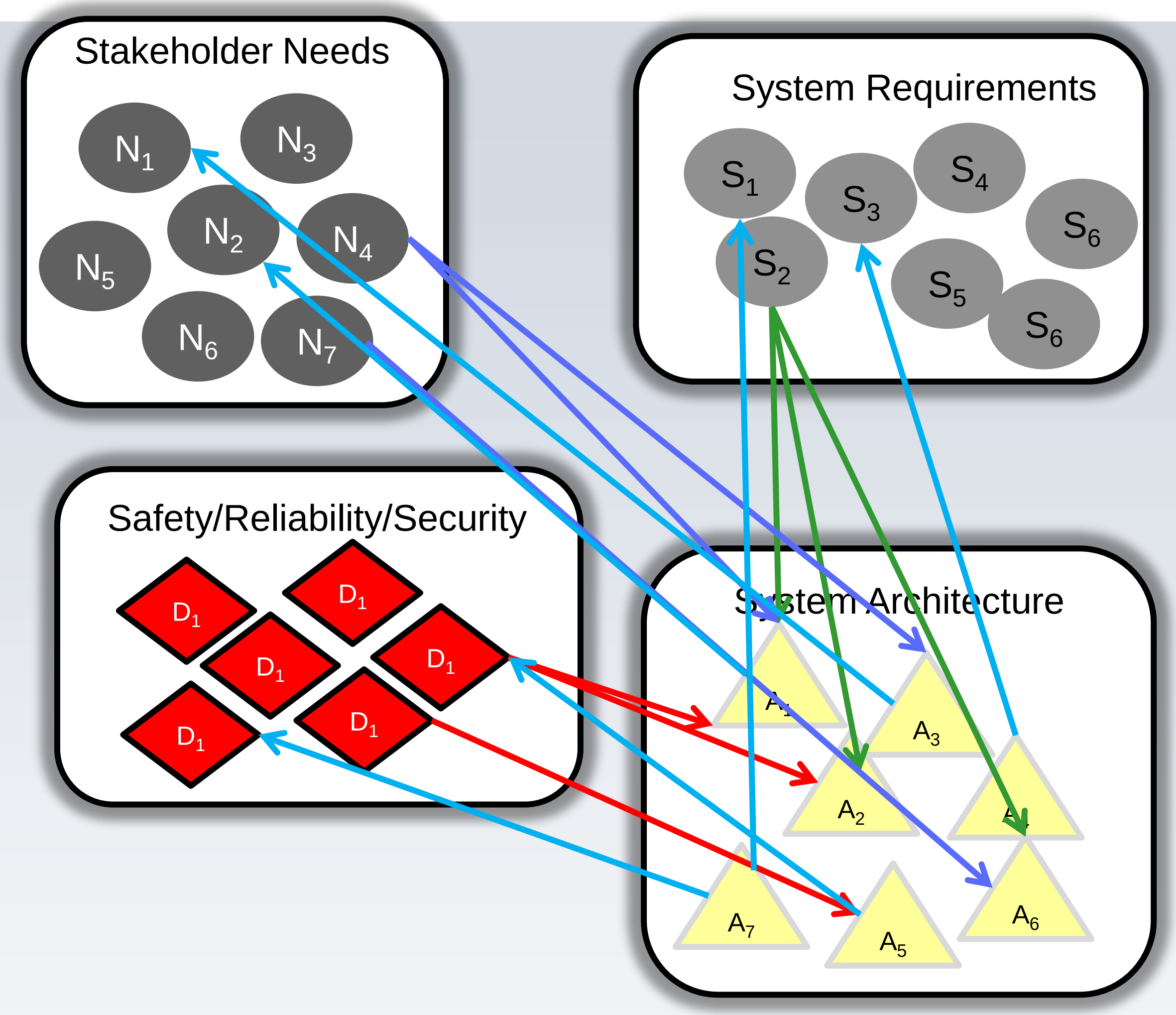
Traceability is data-centric not document-centric



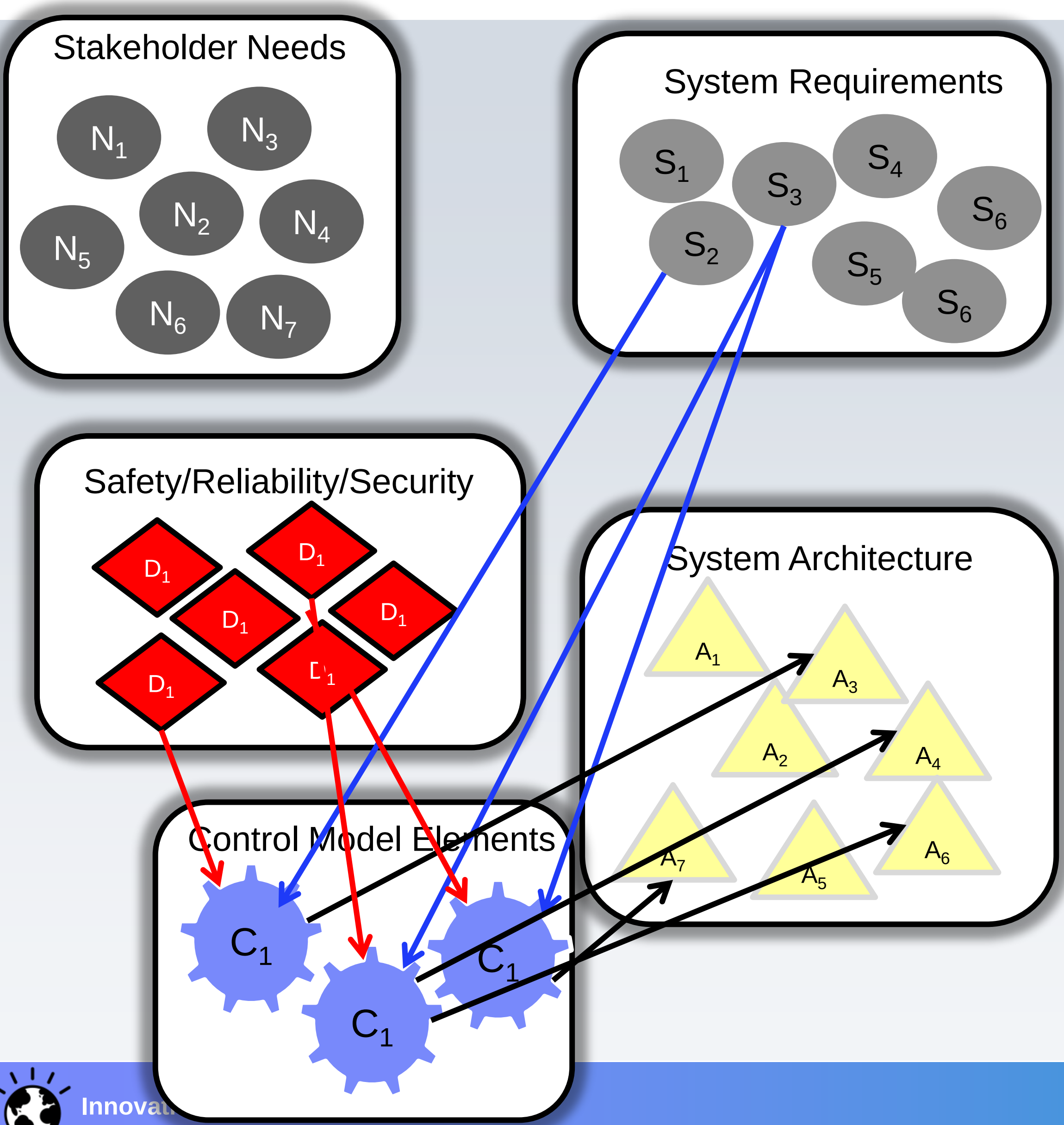
Traceability is data-centric not document-centric



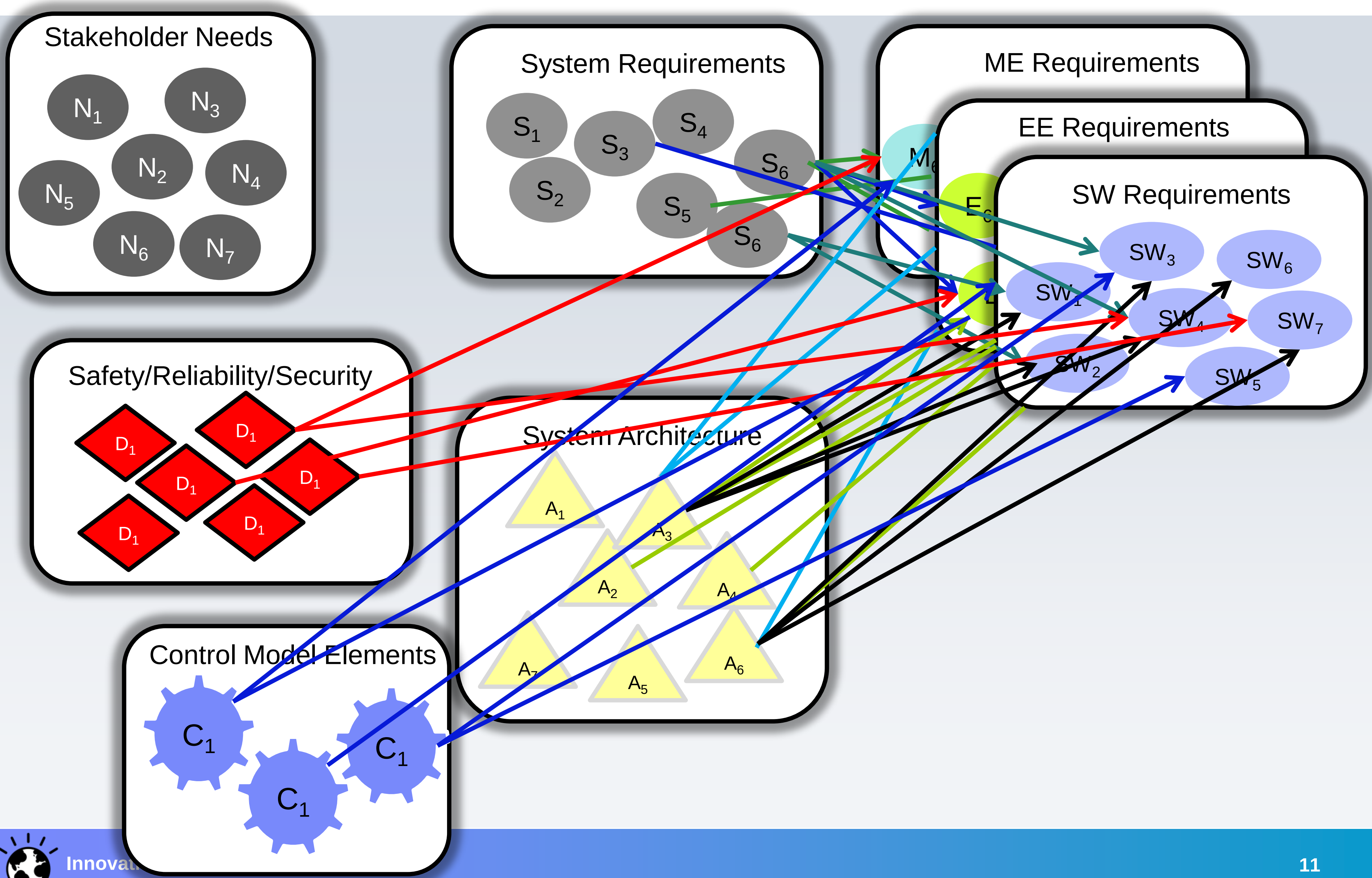
Traceability is data-centric not document-centric



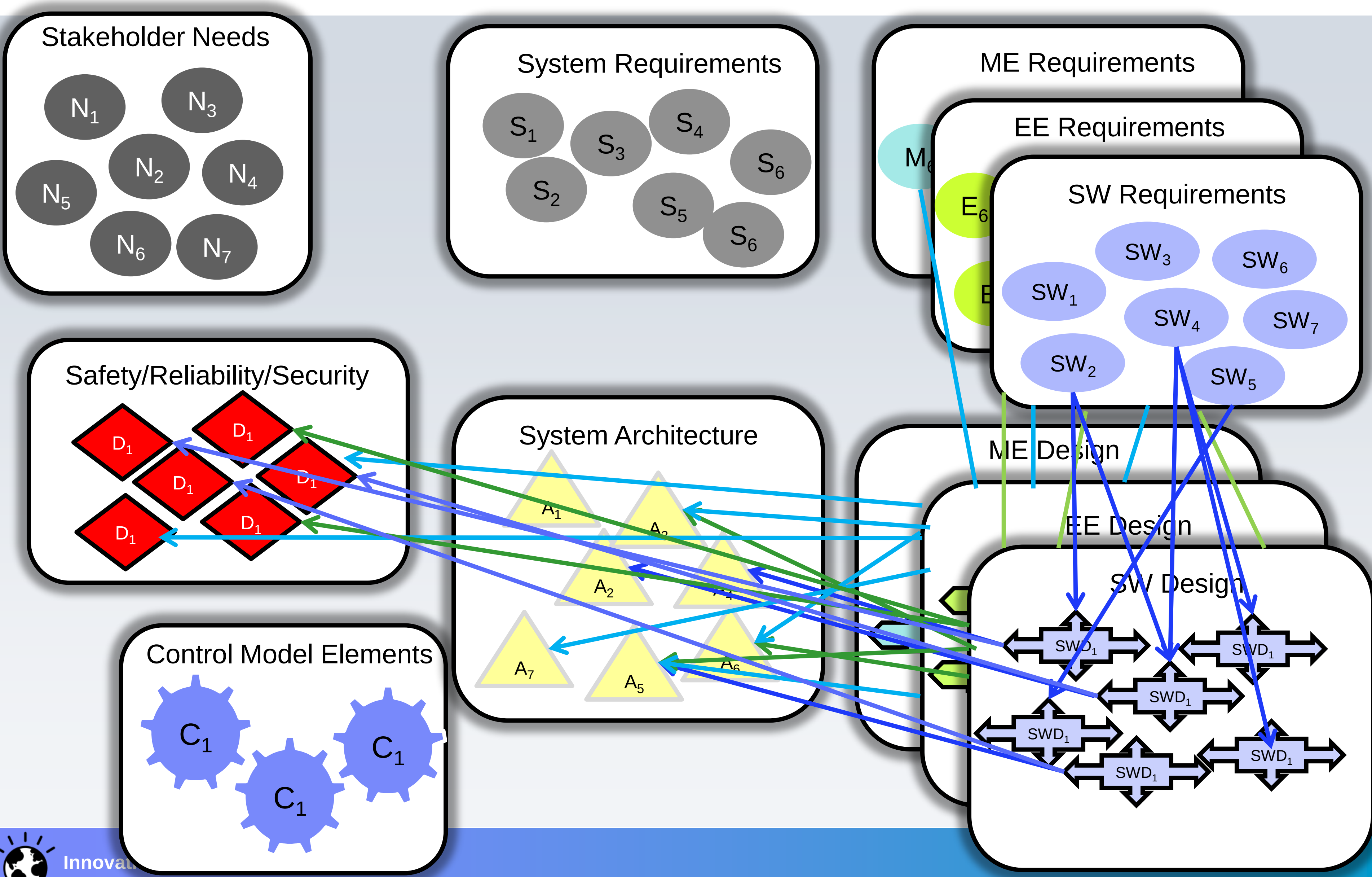
Traceability is data-centric not document-centric



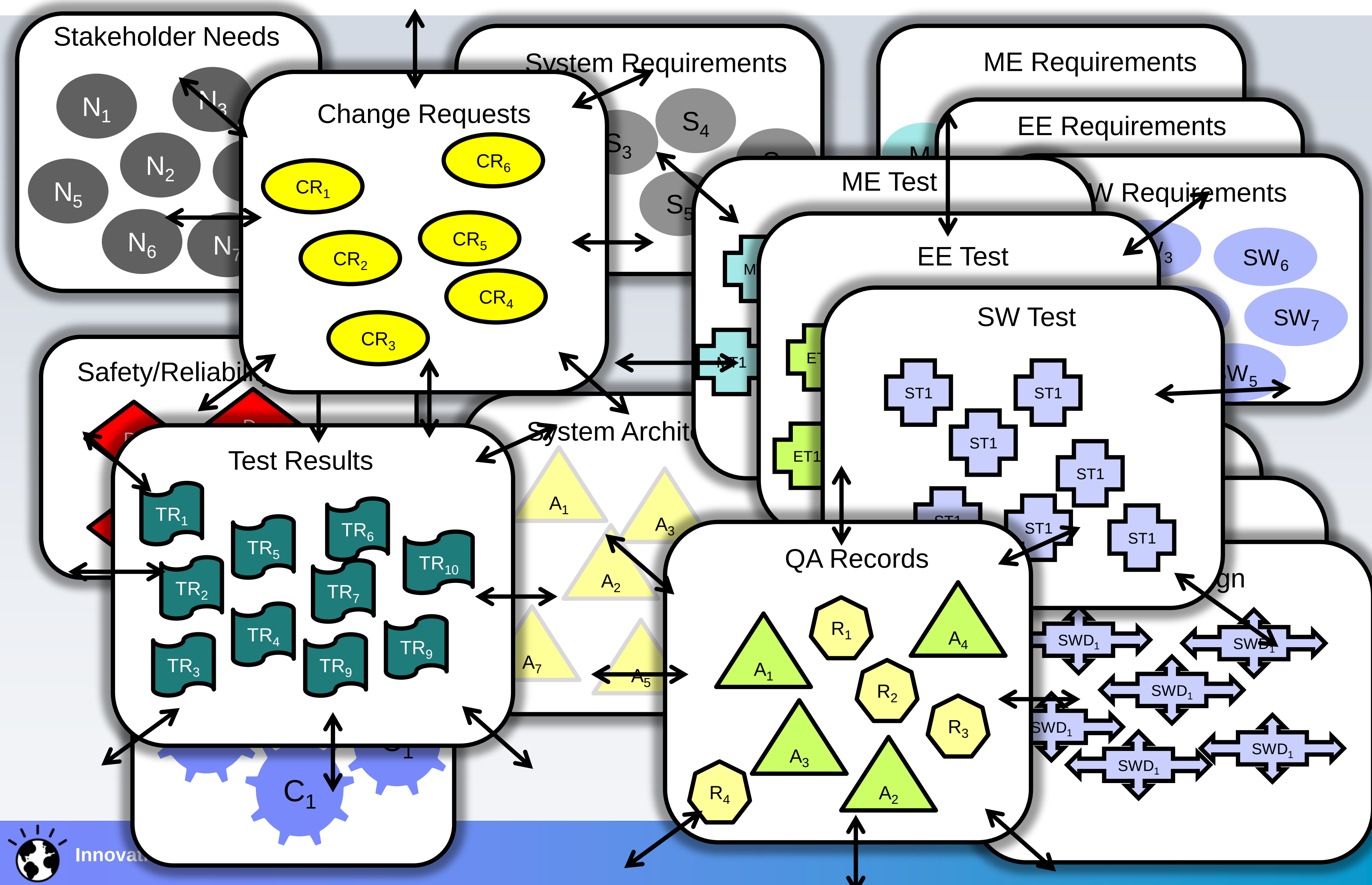
Traceability is data-centric not document-centric



Traceability is data-centric not document-centric



Traceability is data-centric not document-centric



Lifecycle Data Trace Metamatrix

		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	Stakeholder Requirements		R			R					R				o	S	R	
2	System Requirements			R	o	S	R	R			o	R	R	S	S	S	R	R
3	Subsystem Requirements				R	S	R	R	R	S	o	R	R	S	S	S		R
4	SW/EE/ME Requirements					S	R	R	R	S	o	R	o	R	S	S		R
5	Dependability analyses						S	S	S	S	o	S	S	o	S	S	o	S
6	System Architecture							R	R	o		R	R		S	S		R
7	Control Model(s)								R	o	o	R	R	R	S	S		R
8	SW/EE/ME Design[1]									R		R	R	R	S	S		R
9	SW/EE/ME Implementation[1]										o	R	R	R	S	S		R
10	Validation Tests											R					R	
11	System Verification Tests												o	o	S	S		R
12	Component/Integration Tests													o	S	S		R
13	SW/EE/ME Tests														S	S		R
14	QA Records															S	S	S
15	CM Index																S	S
16	Validation Test Results																	
17	Verification Test Results[2]																	

[1] These are usually in separate work products, so this is really multiple items

[2] Tests occur at unit, component, integration, and system levels; again multiple items

Notation	Interpretation
R	Required for reliable, repeatable system development
S	Required for safety critical / high reliability / high security
o	Optional

How Much Traceability?

- The Standards argument
  - Some standard explicitly define the required level of traceability
  - Example:
    - DO-178A required that each requirement was traced to the set of computer op codes that realized it (realization) and that each op code could be shown to be there to at least partially meet a requirement
    - With DO-178B/C, requirements are mapped to a set of source language statements and each source statement must trace back to one or more requirements (provided the set of used source statement → op code are fully verified elsewhere)

How Much Traceability?

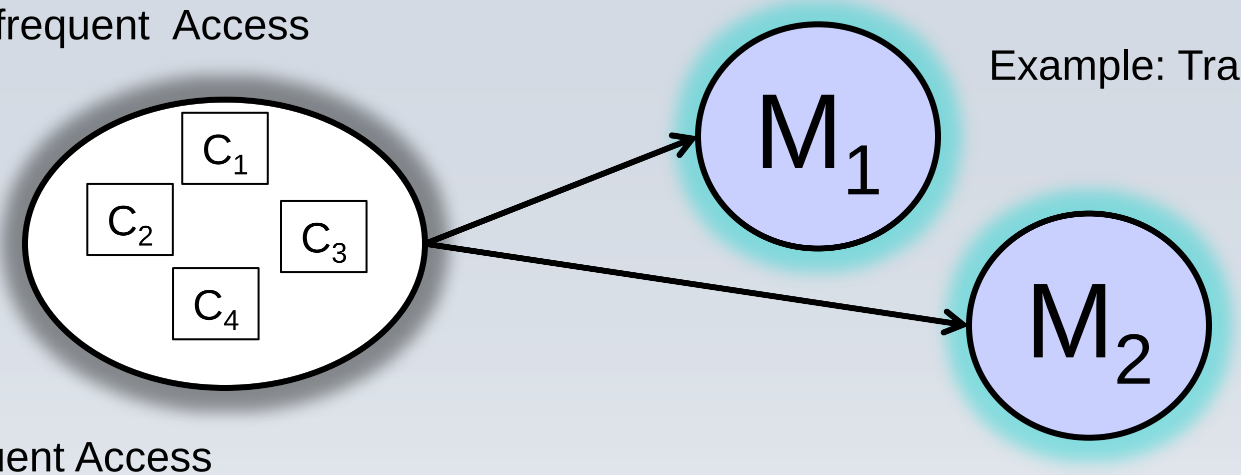
- The cost per probe argument
  - Navigating a trace link has two components: manual searching about (M) after the automated link gets you close to the desired data element
  - Creating and maintaining a trace link has a cost (C) that is incurred when the link is created or modified
  - The total cost for traceability is

$$Total\ Cost = \sum_{total\ links} C + \sum_{Accesses} M$$

  - Two ways to minimize Total Cost:
    - If you will access the links infrequently, then having a larger access cost per query (M) minimizes total cost when you spend less effort creating detailed traceability (lower C)
    - If you will access the links frequently, it makes economic sense to spend more time creating and managing the links (higher C) in order to minimize the time spent manually searching (lower M)

How Much Traceability?

Case 1: Infrequent Access



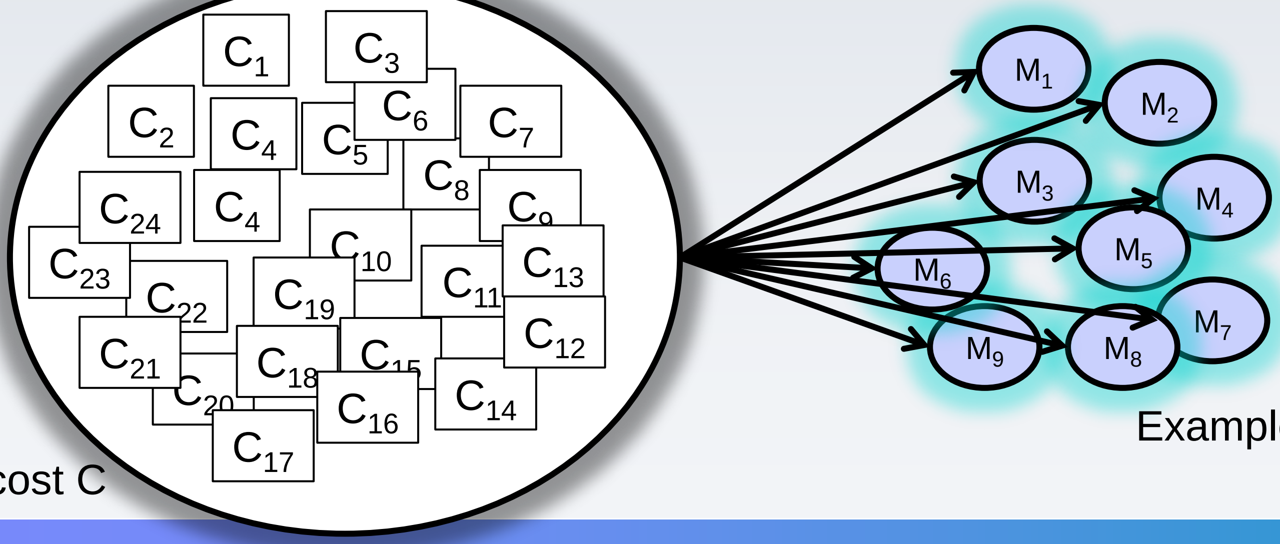
Low cost C

Example: Trace to source file

High cost M

Case 2: Frequent Access

Why? Frequent requirements change; high defect density (so lots of maintenance)



Low cost M

Example: Trace to source line(s)

High cost C

Common Levels of Traceability

- Coarse (Low C, very high M)
  - Map work products
    - Requirement Spec → Design Document
    - Design Document → Directory of Source Files
    - Requirements Spec → Test Plan → Test Suite

Common Levels of Traceability

- Medium (Moderate C, Moderate M)
  - Map large scale elements
    - Requirement (*) → Use case (1)
    - Use Case (1) → Large design elements (*) || Design Diagram (1)
    - Design Element (1) -> Class, Function, or Data Structure (*)
    - Requirement (1) → Test Case (*)

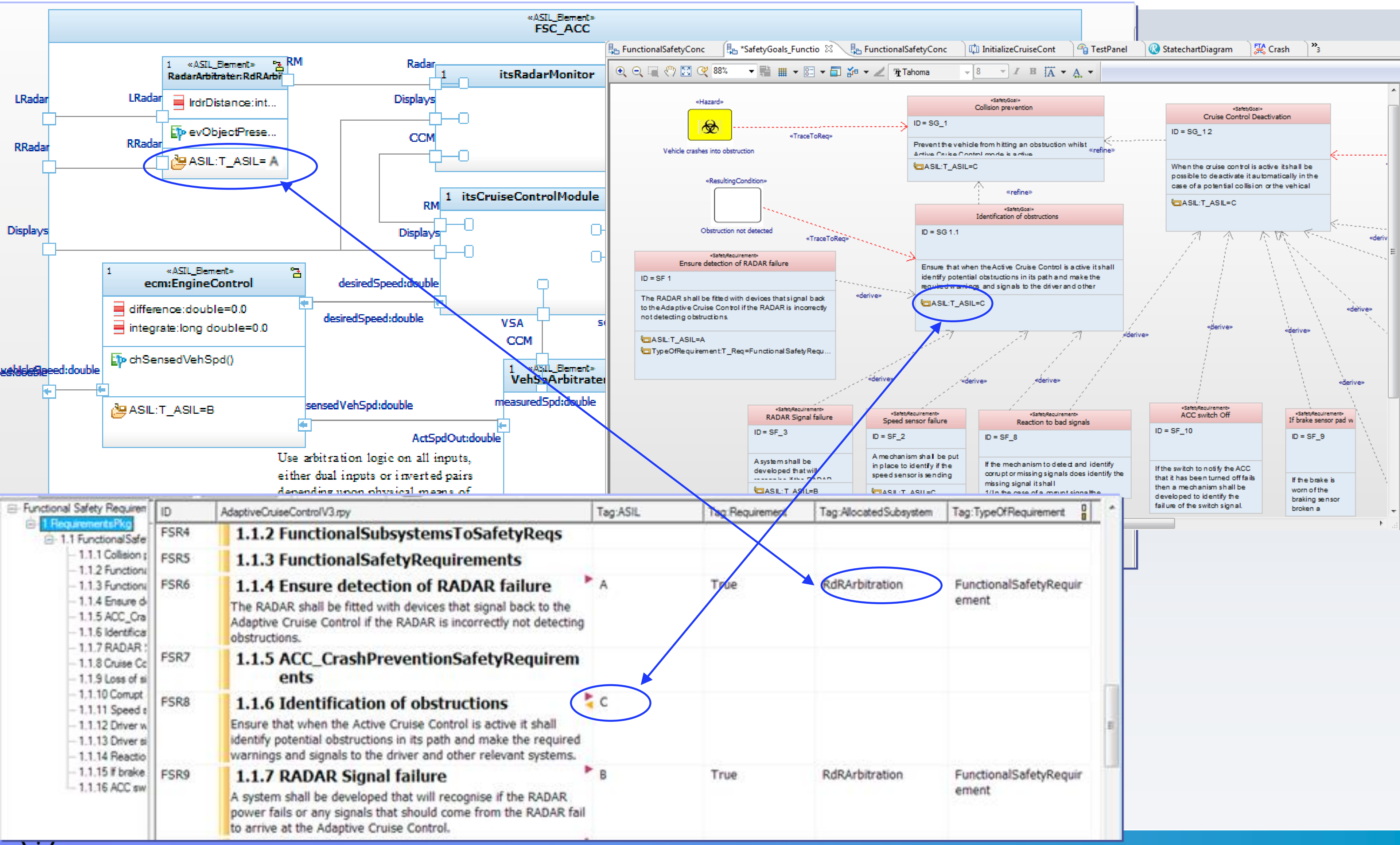
Common Levels of Traceability

- Medium High (Moderate C, Moderate M, Safety Criticality Moderate)
  - Map large scale elements
    - Requirement (*) → Use case (1)
    - Requirement (*) → System function (use case operation/event reception, flow property)(*)
    - Use Case (1) → Large design elements (*) || Design Diagram (1)
    - Design Element (1) -> Class, Function, or Data Structure (*)
    - Requirement (1) → Test Case (*)

Common Levels of Traceability

- Fine (High C, low M, Safety Criticality High)
  - Map to small scale elements
    - Requirement (*) → Use case (1)
    - Requirement (*) → Design Element (*)
    - Requirement (*) → Class or Function in source code (*) (or smaller)
    - Requirement (1) → Test Case(*)
    - Test Case (*) → Design Element (1)
    - Test Case (*) → Class or Function in source code (*) (or smaller)

What does traceability look like?



Use arbitration logic on all inputs, either dual inputs or inverted pairs depending upon physical means of

«SafetyGoal» Collision prevention
ID = SG_1
Prevent the vehicle from hitting an obstruction whilst Active Cruise Control mode is active
ASIL:T_ASIL=C

«SafetyGoal» Cruise Control Deactivation
ID = SG_12
When the cruise control is active it shall be possible to deactivate it automatically in the case of a potential collision or the vehical
ASIL:T_ASIL=C

«SafetyGoal» Identification of obstructions
ID = SG 1.1
Ensure that when the Active Cruise Control is active it shall identify potential obstructions in its path and make the required warnings and signals to the driver and other
ASIL:T_ASIL=C

«SafetyRequirement» Ensure detection of RADAR failure
ID = SF 1
The RADAR shall be fitted with devices that signal back to the Adaptive Cruise Control if the RADAR is incorrectly not detecting obstructions.
ASIL:T_ASIL=A
TypeOfRequirement:T_Req=FunctionalSafetyRequ...

«SafetyRequirement» RADAR Signal failure
ID = SF_3
A system shall be developed that will
ASIL:T_ASIL=B

«SafetyRequirement» Speed sensor failure
ID = SF_2
A mechanism shall be put in place to identify if the speed sensor is sending
ASIL:T_ASIL=C

«SafetyRequirement» Reaction to bad signals
ID = SF_8
If the mechanism to detect and identify corrupt or missing signals does identify the missing signal it shall
1/In the case of a corrupt signal the

«SafetyRequirement» ACC switch Off
ID = SF_10
If the switch to notify the ACC that it has been turned off fails then a mechanism shall be developed to identify the failure of the switch signal.

«SafetyRequirement» If brake sensor pad w
ID = SF_9
If the brake is worn of the braking sensor broken a

ID	AdaptiveCruiseControlV3.rpy	Tag:ASIL	Tag:Requirement	Tag:AllocatedSubsystem	Tag:TypeOfRequirement
FSR4	**1.1.2 FunctionalSubsystemsToSafetyReqs**				
FSR5	**1.1.3 FunctionalSafetyRequirements**				
FSR6	**1.1.4 Ensure detection of RADAR failure** The RADAR shall be fitted with devices that signal back to the Adaptive Cruise Control if the RADAR is incorrectly not detecting obstructions.	A	True	RdRArbitration	FunctionalSafetyRequirement
FSR7	**1.1.5 ACC_CrashPreventionSafetyRequirements**				
FSR8	**1.1.6 Identification of obstructions** Ensure that when the Active Cruise Control is active it shall identify potential obstructions in its path and make the required warnings and signals to the driver and other relevant systems.	C			
FSR9	**1.1.7 RADAR Signal failure** A system shall be developed that will recognise if the RADAR power fails or any signals that should come from the RADAR fail to arrive at the Adaptive Cruise Control.	B	True	RdRArbitration	FunctionalSafetyRequirement

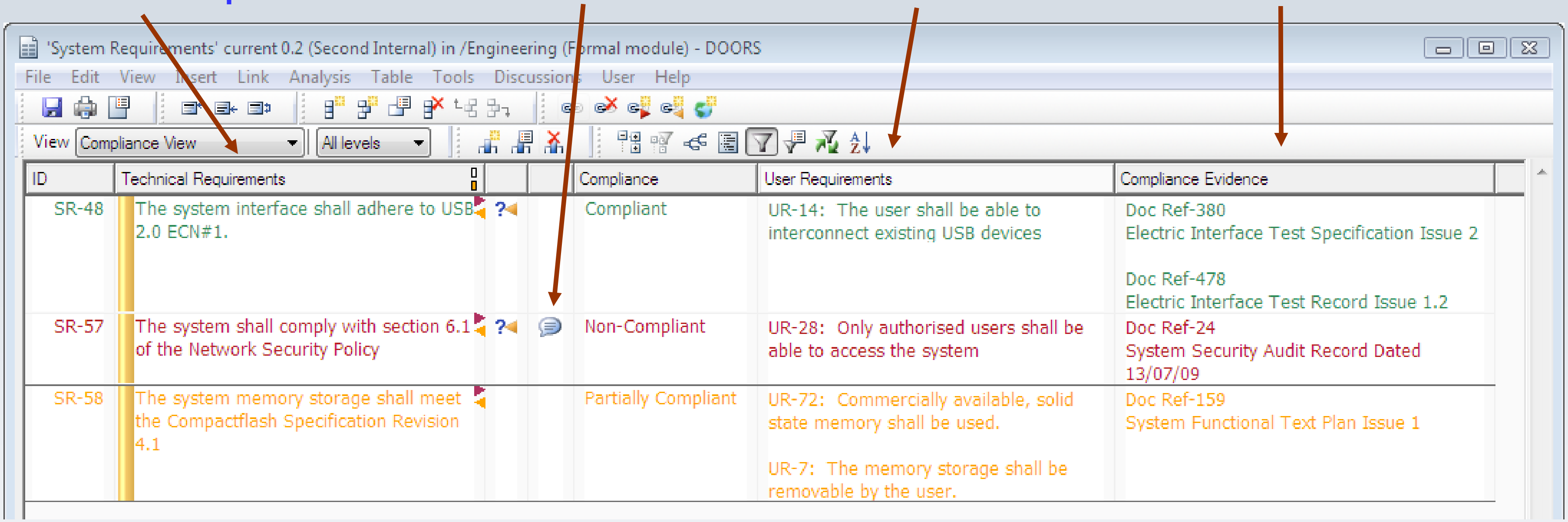
What does traceability look like?

Technical Reqs

Discussions

Higher Level Reqs

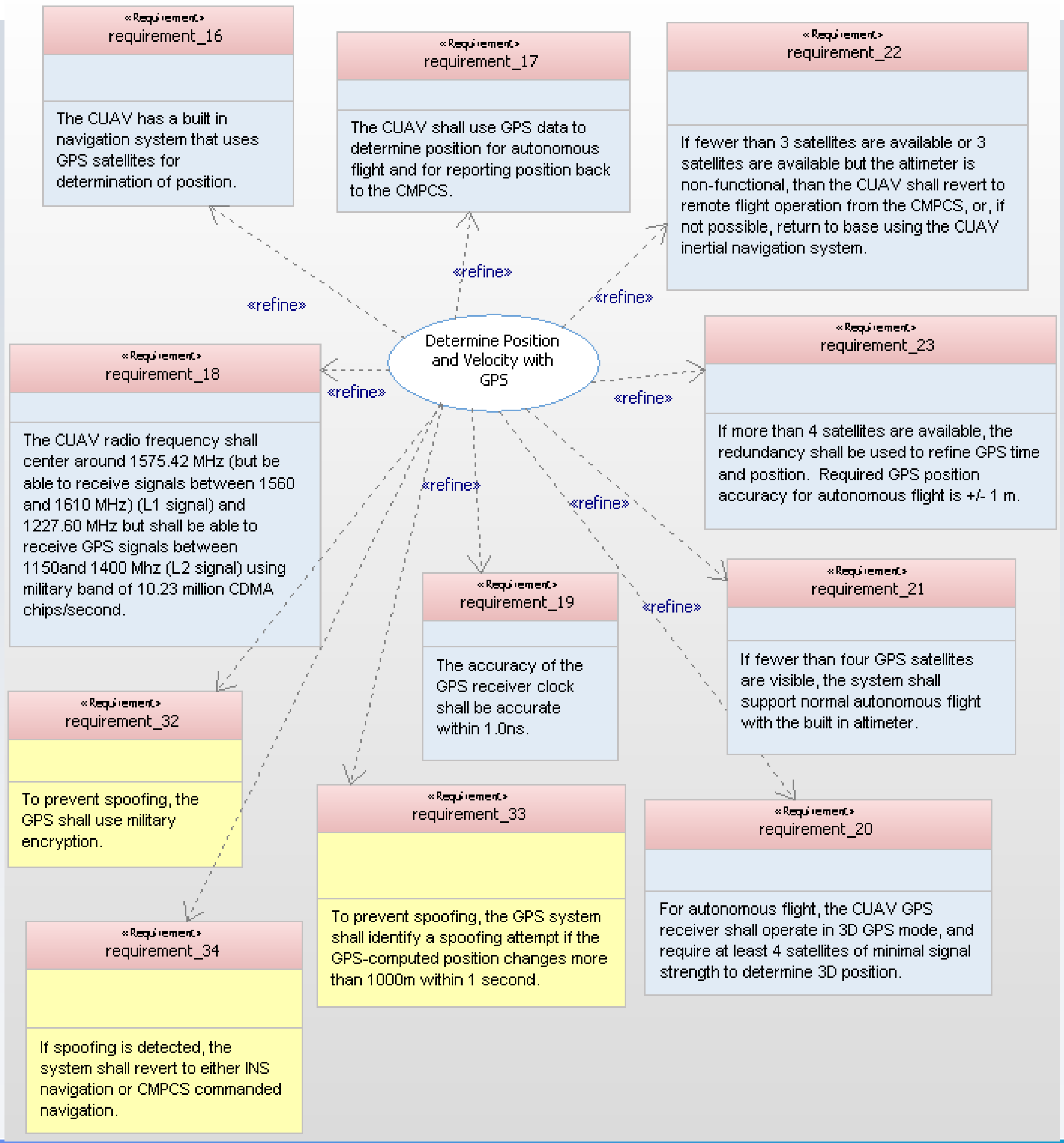
Evidence Detail


'System Requirements' current 0.2 (Second Internal) in /Engineering (Formal module) - DOORS
File Edit View Insert Link Analysis Table Tools Discussions User Help
View Compliance View
All levels

ID	Technical Requirements			Compliance	User Requirements	Compliance Evidence
SR-48	The system interface shall adhere to USB 2.0 ECN#1.	?		Compliant	UR-14: The user shall be able to interconnect existing USB devices	Doc Ref-380 Electric Interface Test Specification Issue 2 Doc Ref-478 Electric Interface Test Record Issue 1.2
SR-57	The system shall comply with section 6.1 of the Network Security Policy	?	💬	Non-Compliant	UR-28: Only authorised users shall be able to access the system	Doc Ref-24 System Security Audit Record Dated 13/07/09
SR-58	The system memory storage shall meet the Compactflash Specification Revision 4.1			Partially Compliant	UR-72: Commercially available, solid state memory shall be used. UR-7: The memory storage shall be removable by the user.	Doc Ref-159 System Functional Text Plan Issue 1

Traceability in Models

- Traceability is modeled in UML with the dependency relation «trace»
- SysML adds specialized kinds
  - «satisfy»
  - «realize»
  - «verify»



«Requirement»
requirement_16
The CUAV has a built in navigation system that uses GPS satellites for determination of position.

«Requirement»
requirement_17
The CUAV shall use GPS data to determine position for autonomous flight and for reporting position back to the CMPCS.

«Requirement»
requirement_22
If fewer than 3 satellites are available or 3 satellites are available but the altimeter is non-functional, than the CUAV shall revert to remote flight operation from the CMPCS, or, if not possible, return to base using the CUAV inertial navigation system.

«Requirement»
requirement_18
The CUAV radio frequency shall center around 1575.42 MHz (but be able to receive signals between 1560 and 1610 MHz) (L1 signal) and 1227.60 MHz but shall be able to receive GPS signals between 1150and 1400 Mhz (L2 signal) using military band of 10.23 million CDMA chips/second.

«Requirement»
requirement_23
If more than 4 satellites are available, the redundancy shall be used to refine GPS time and position. Required GPS position accuracy for autonomous flight is +/- 1 m.

«Requirement»
requirement_19
The accuracy of the GPS receiver clock shall be accurate within 1.0ns.

«Requirement»
requirement_21
If fewer than four GPS satellites are visible, the system shall support normal autonomous flight with the built in altimeter.

«Requirement»
requirement_32
To prevent spoofing, the GPS shall use military encryption.

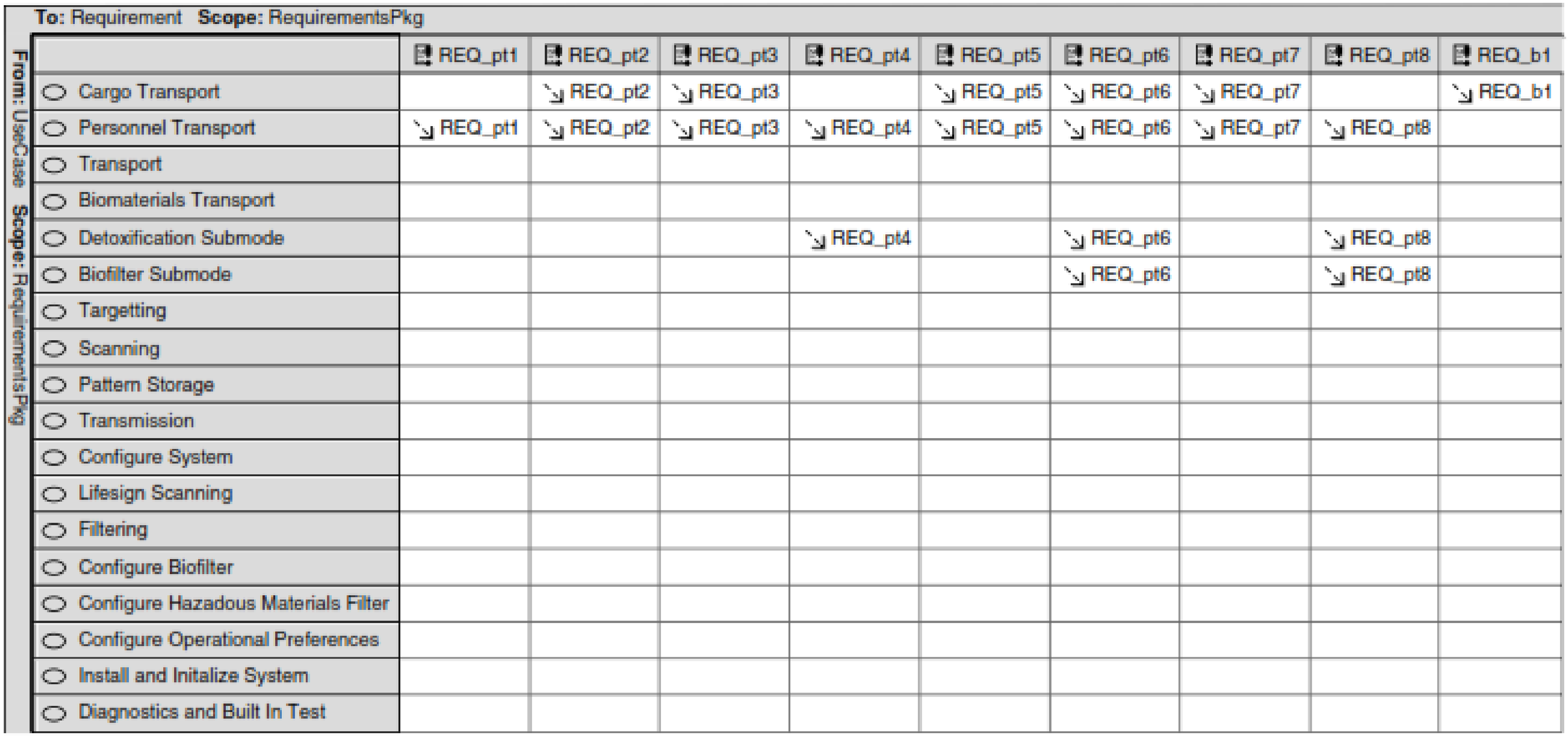
«Requirement»
requirement_33
To prevent spoofing, the GPS system shall identify a spoofing attempt if the GPS-computed position changes more than 1000m within 1 second.

«Requirement»
requirement_20
For autonomous flight, the CUAV GPS receiver shall operate in 3D GPS mode, and require at least 4 satellites of minimal signal strength to determine 3D position.

«Requirement»
requirement_34
If spoofing is detected, the system shall revert to either INS navigation or CMPCS commanded navigation.

Traceability in Models

- Trace links can be shown in tabular form in Rational Rhapsody as well



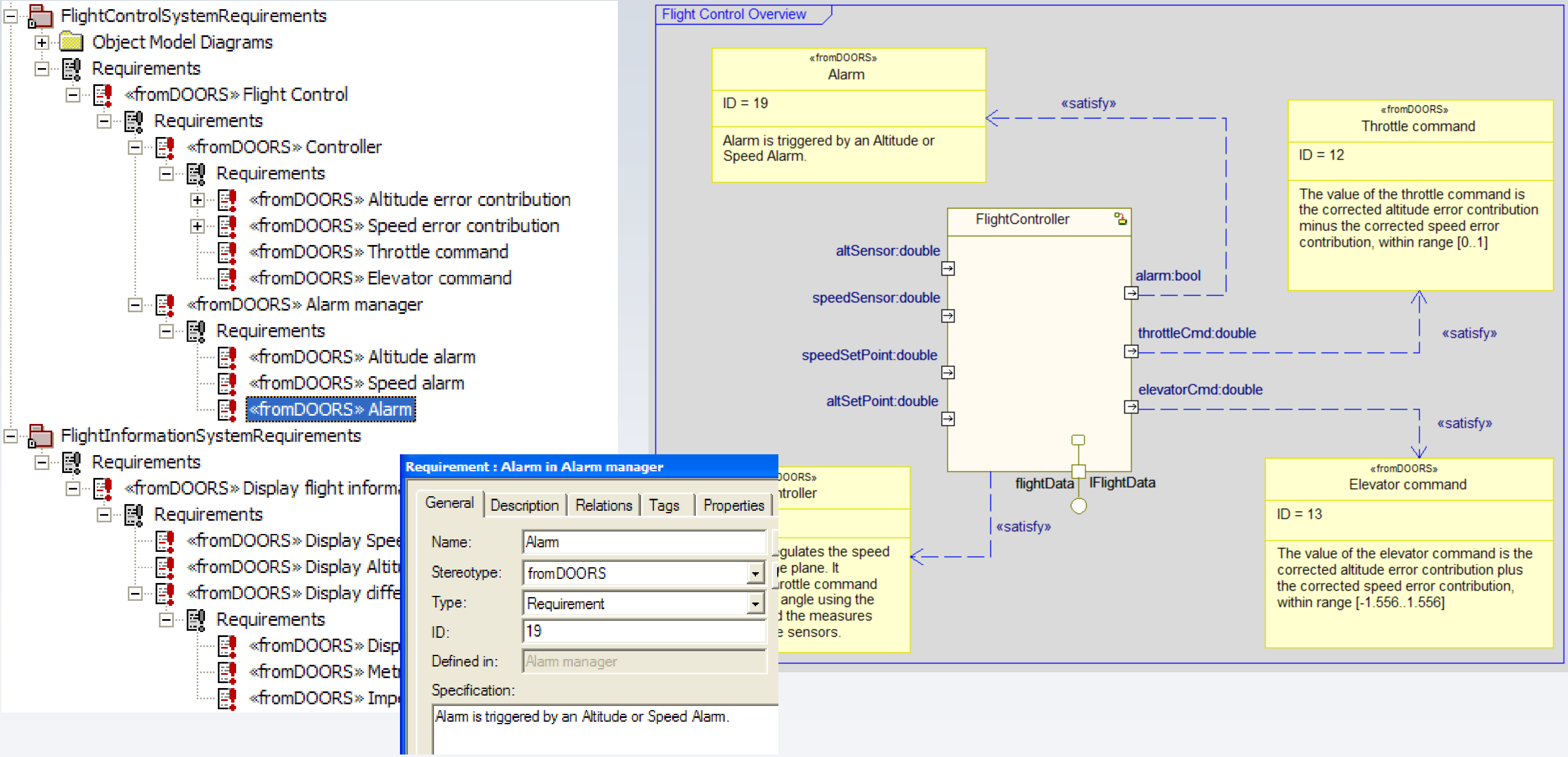
To: Requirement **Scope:** RequirementsPkg

From: UseCase **Scope:** RequirementsPkg	REQ_pt1	REQ_pt2	REQ_pt3	REQ_pt4	REQ_pt5	REQ_pt6	REQ_pt7	REQ_pt8	REQ_b1
○ Cargo Transport		↘ REQ_pt2	↘ REQ_pt3		↘ REQ_pt5	↘ REQ_pt6	↘ REQ_pt7		↘ REQ_b1
○ Personnel Transport	↘ REQ_pt1	↘ REQ_pt2	↘ REQ_pt3	↘ REQ_pt4	↘ REQ_pt5	↘ REQ_pt6	↘ REQ_pt7	↘ REQ_pt8	
○ Transport									
○ Biomaterials Transport									
○ Detoxification Submode				↘ REQ_pt4		↘ REQ_pt6		↘ REQ_pt8	
○ Biofilter Submode						↘ REQ_pt6		↘ REQ_pt8	
○ Targetting									
○ Scanning									
○ Pattern Storage									
○ Transmission									
○ Configure System									
○ Lifesign Scanning									
○ Filtering									
○ Configure Biofilter									
○ Configure Hazadous Materials Filter									
○ Configure Operational Preferences									
○ Install and Initalize System									
○ Diagnostics and Built In Test									

Figure 6.19 *Requirements Traceability Matrix*

Rational DOORS is closely linked with Rational Rhapsody

- Rational Rhapsody can import or reference requirements from Rational DOORS
- Trace links in Rhapsody can be pushed out to DOORS



FlightControlSystemRequirements
- Object Model Diagrams
- Requirements
  - «fromDOORS» Flight Control
    - Requirements
      - «fromDOORS» Controller
        - Requirements
          - «fromDOORS» Altitude error contribution
          - «fromDOORS» Speed error contribution
          - «fromDOORS» Throttle command
          - «fromDOORS» Elevator command
      - «fromDOORS» Alarm manager
        - Requirements
          - «fromDOORS» Altitude alarm
          - «fromDOORS» Speed alarm
          - «fromDOORS» Alarm

FlightInformationSystemRequirements
- Requirements
  - «fromDOORS» Display flight informa
    - Requirements
      - «fromDOORS» Display Spee
      - «fromDOORS» Display Altit
      - «fromDOORS» Display diffe
        - Requirements
          - «fromDOORS» Disp
          - «fromDOORS» Met
          - «fromDOORS» Impe

Requirement : Alarm in Alarm manager

General | Description | Relations | Tags | Properties

Name: Alarm

Stereotype: fromDOORS

Type: Requirement

ID: 19

Defined in: Alarm manager

Specification:

Alarm is triggered by an Altitude or Speed Alarm.

Flight Control Overview

«fromDOORS»
Alarm
ID = 19
Alarm is triggered by an Altitude or Speed Alarm.

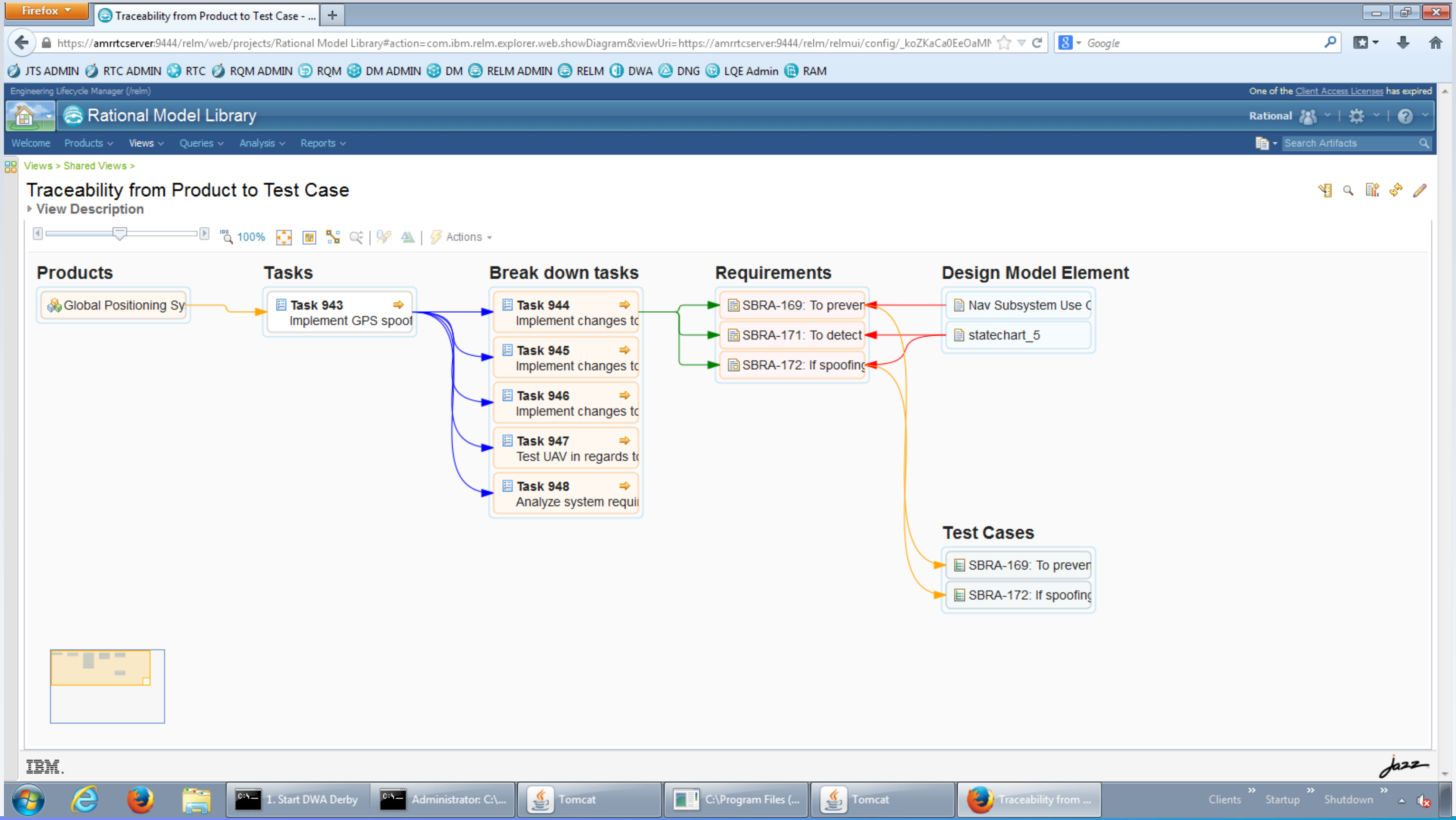
«fromDOORS»
Throttle command
ID = 12
The value of the throttle command is the corrected altitude error contribution minus the corrected speed error contribution, within range [0..1]

«fromDOORS»
Elevator command
ID = 13
The value of the elevator command is the corrected altitude error contribution plus the corrected speed error contribution, within range [-1.556..1.556]

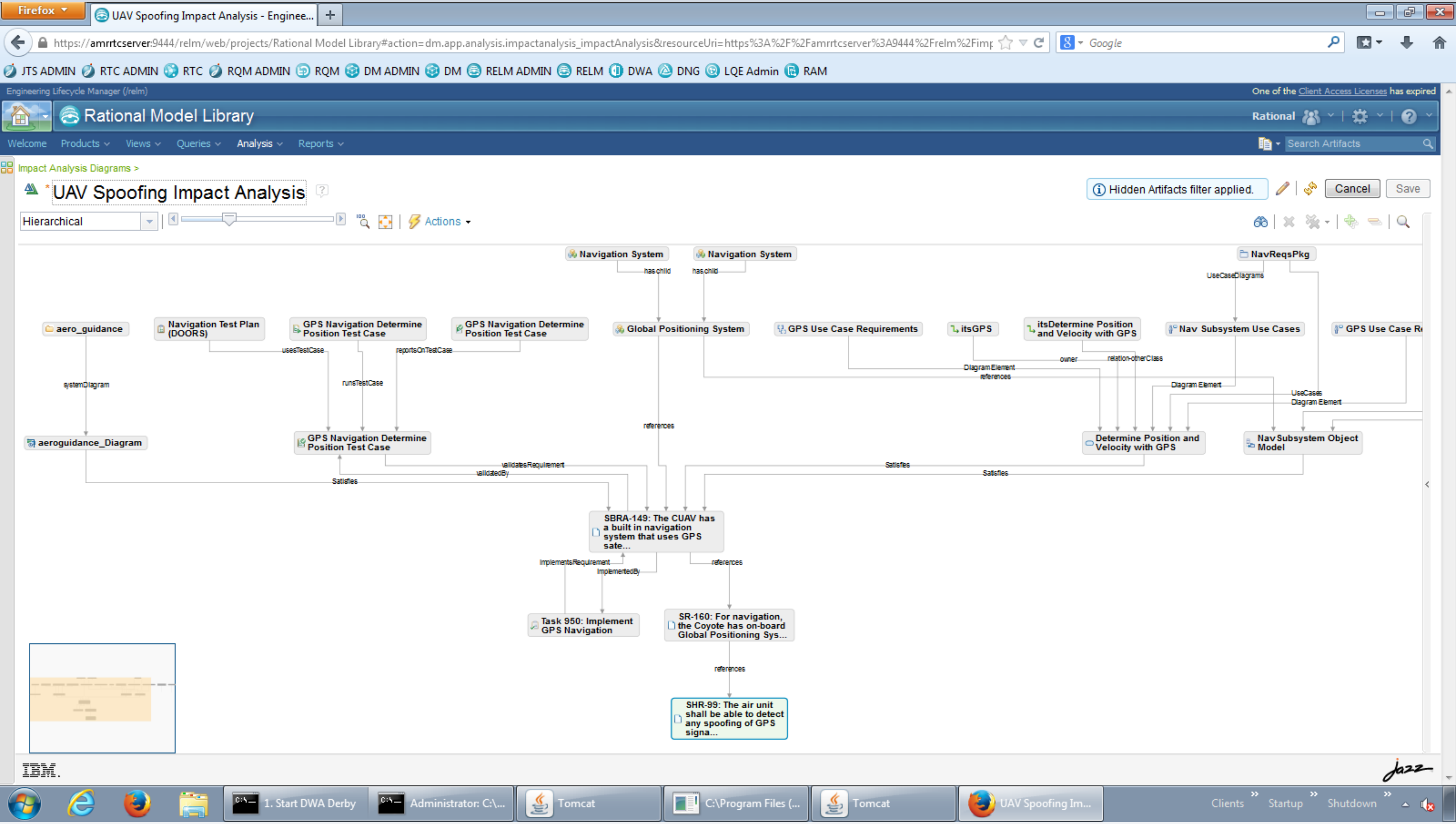
OORS»
troller
gulates the speed
e plane. It
rottle command
angle using the
the measures
e sensors.

Rational Engineering Lifecycle Manager (RELM)

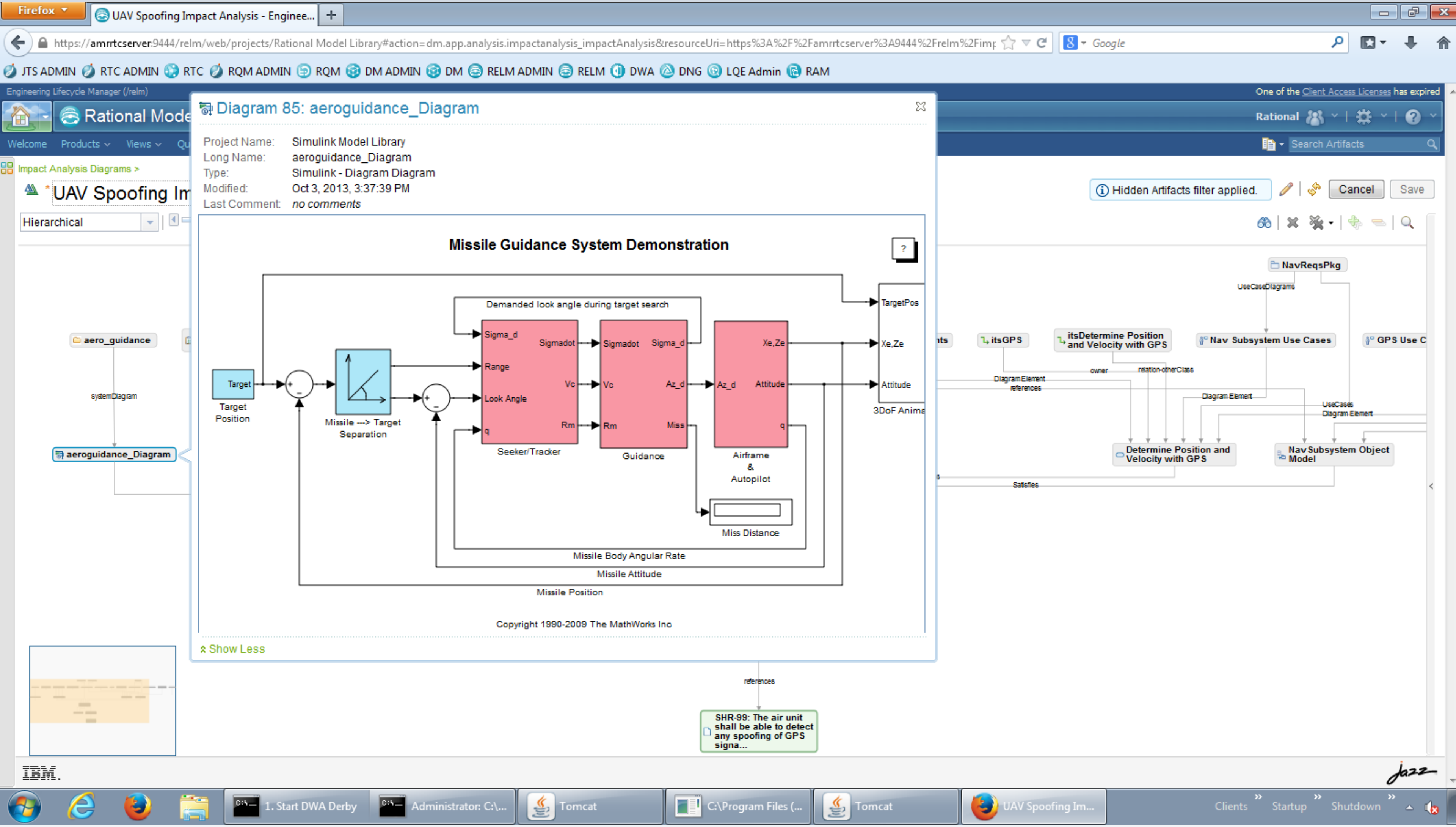
- RELM is a tool meant to aggregate linked engineering data from a variety of sources to support traceability, impact analysis, etc. including both IBM and non-IBM tools with pre-defined and user-defined views



Traceability with RELM


Firefox
UAV Spoofing Impact Analysis - Enginee...
https://amrrtcserver:9444/relm/web/projects/Rational Model Library#action=dm.app.analysis.impactanalysis_impactAnalysis&resourceUri=https%3A%2F%2Famrrtcserver%3A9444%2Frelm%2Fimp
Google
JTS ADMIN
RTC ADMIN
RTC
RQM ADMIN
RQM
DM ADMIN
DM
RELM ADMIN
RELM
DWA
DNG
LQE Admin
RAM
Engineering Lifecycle Manager (/relm)
One of the Client Access Licenses has expired
Rational Model Library
Rational
Welcome
Products
Views
Queries
Analysis
Reports
Search Artifacts
Impact Analysis Diagrams >
*UAV Spoofing Impact Analysis
Hidden Artifacts filter applied.
Cancel
Save
Hierarchical
Actions
Navigation System
Navigation System
NavReqsPkg
has child
has child
UseCaseDiagrams
aero_guidance
Navigation Test Plan (DOORS)
GPS Navigation Determine Position Test Case
GPS Navigation Determine Position Test Case
Global Positioning System
GPS Use Case Requirements
itsGPS
itsDetermine Position and Velocity with GPS
Nav Subsystem Use Cases
GPS Use Case Re
usesTestCase
reportsOnTestCase
owner
relation-otherClass
DiagramElement references
systemDiagram
runsTestCase
Diagram Element
UseCases
Diagram Element
references
aeroguidance_Diagram
GPS Navigation Determine Position Test Case
Determine Position and Velocity with GPS
Nav Subsystem Object Model
validatesRequirement
validatedBy
Satisfies
Satisfies
Satisfies
SBRA-149: The CUAV has a built in navigation system that uses GPS sate...
ImplementsRequirement
ImplementedBy
references
Task 950: Implement GPS Navigation
SR-160: For navigation, the Coyote has on-board Global Positioning Sys...
references
SHR-99: The air unit shall be able to detect any spoofing of GPS signa...
IBM.
Jazz
1. Start DWA Derby
Administrator: C:\...
Tomcat
C:\Program Files (...
Tomcat
UAV Spoofing Im...
Clients
Startup
Shutdown

Traceability with RELM



https://amrrtcserver:9444/relm/web/projects/Rational Model Library#action=dm.app.analysis.impactanalysis_impactAnalysis&resourceUri=https%3A%2F%2Famrrtcserver%3A9444%2Frelm%2Fimp

Diagram 85: aeroguidance_Diagram

Project Name:	Simulink Model Library
Long Name:	aeroguidance_Diagram
Type:	Simulink - Diagram Diagram
Modified:	Oct 3, 2013, 3:37:39 PM
Last Comment:	*no comments*

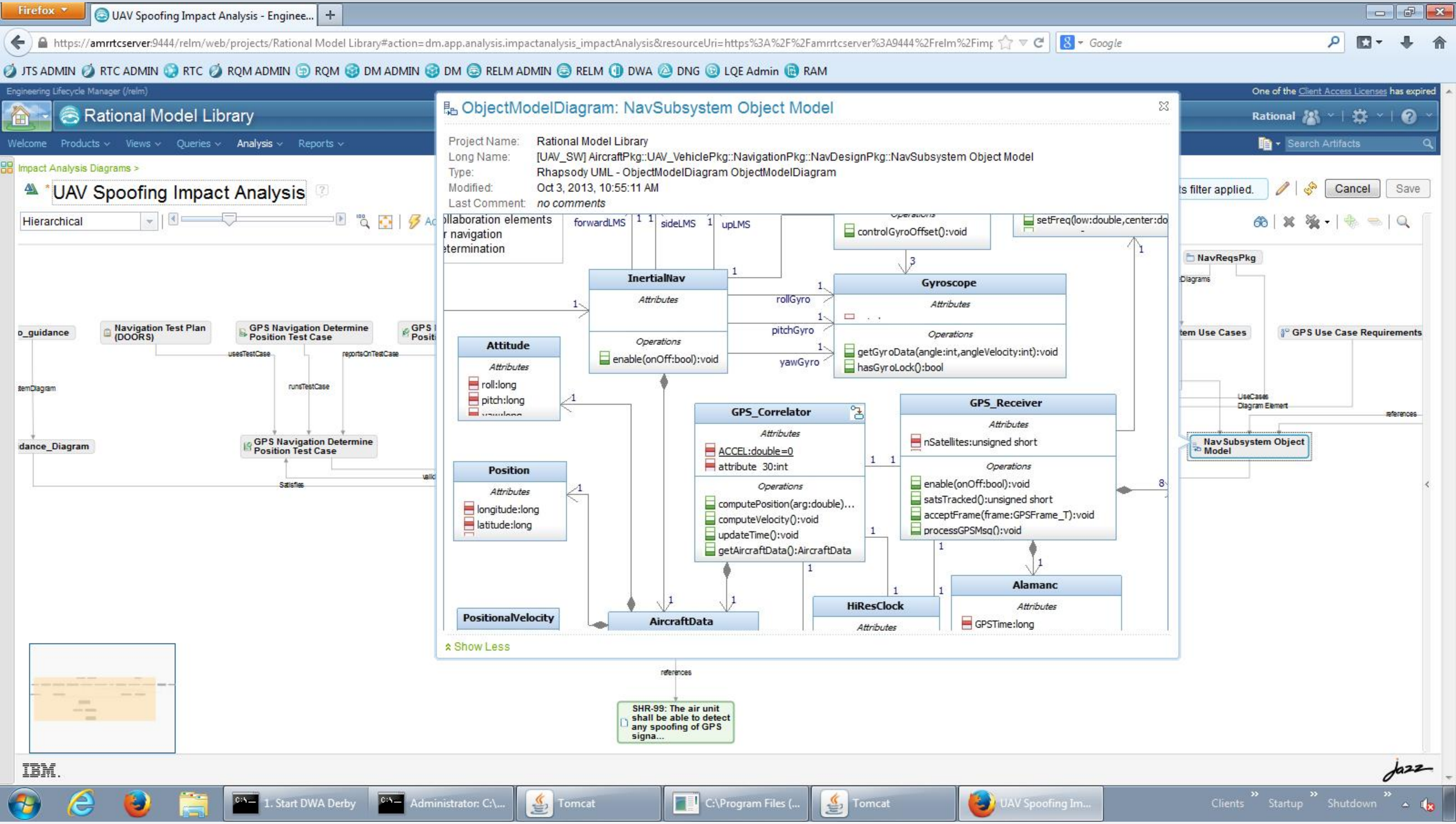
Copyright 1990-2009 The MathWorks Inc

SHR-99: The air unit shall be able to detect any spoofing of GPS signa...

One of the Client Access Licenses has expired

Hidden Artifacts filter applied.

Traceability with RELM

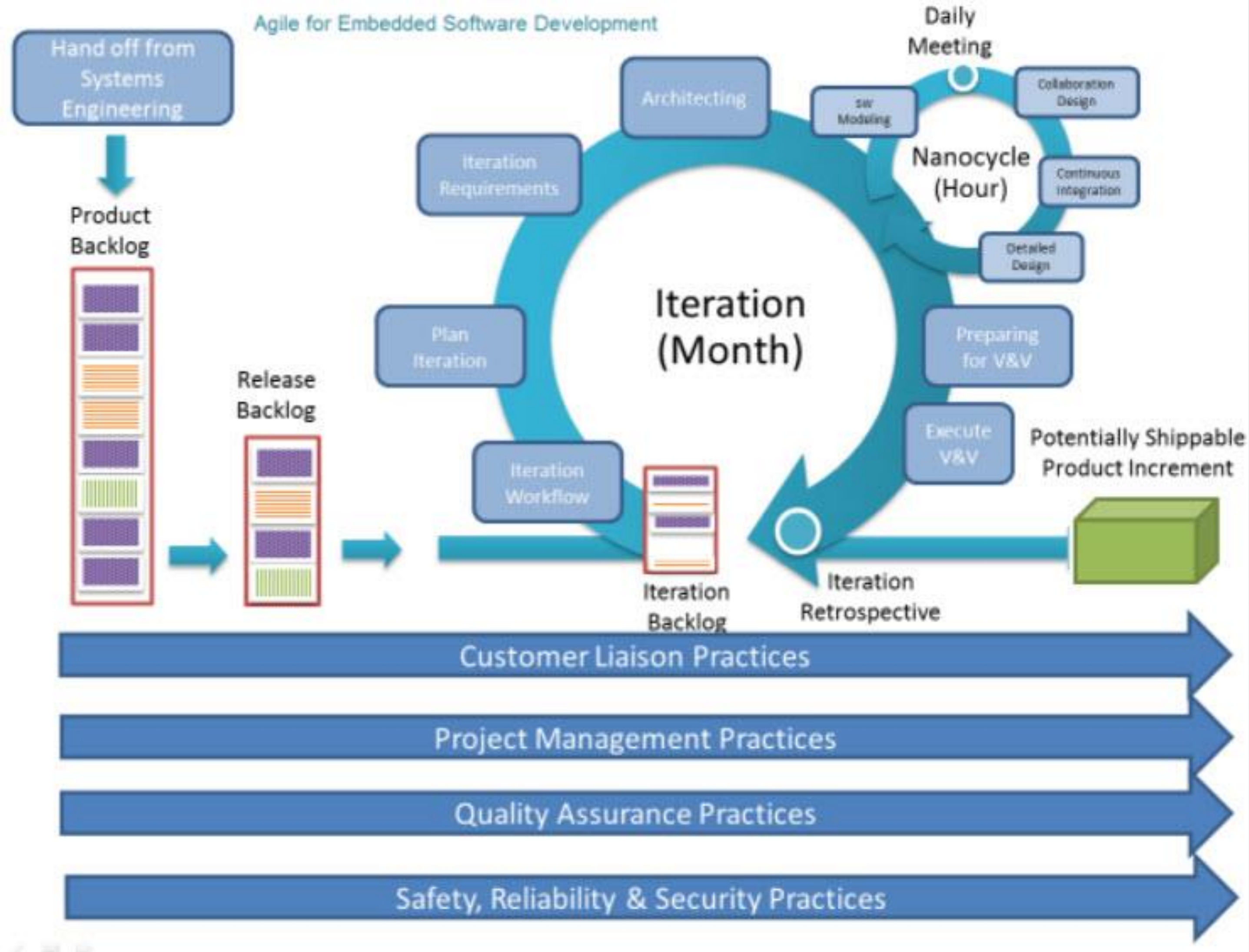


ObjectModelDiagram: NavSubsystem Object Model

Project Name: Rational Model Library
Long Name: [UAV_SW] AircraftPkg::UAV_VehiclePkg::NavigationPkg::NavDesignPkg::NavSubsystem Object Model
Type: Rhapsody UML - ObjectModelDiagram ObjectModelDiagram
Modified: Oct 3, 2013, 10:55:11 AM
Last Comment: *no comments*

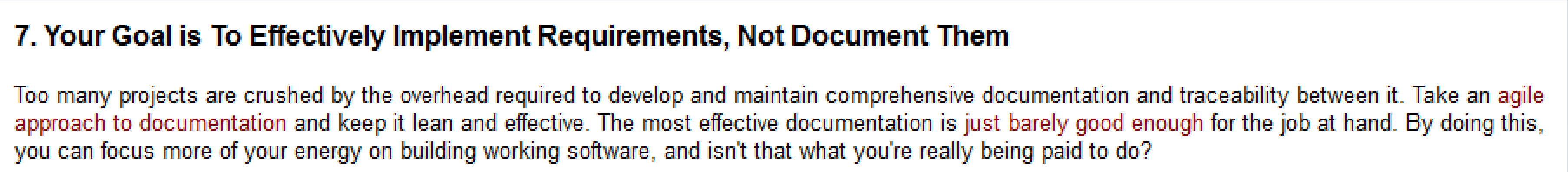
SHR-99: The air unit shall be able to detect any spoofing of GPS signa...

Fitting Traceability into the Agile Lifestyle



What does the Agile literature say about traceability?

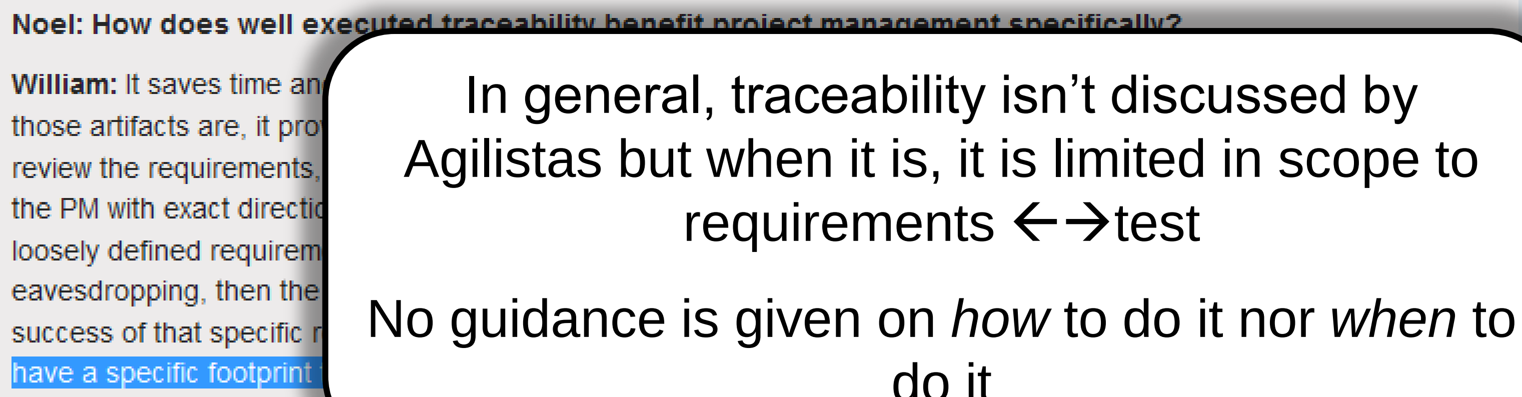
Scott Ambler (http://www.agilemodeling.com/essays/agileRequirementsBestPractices.htm)



7. Your Goal is To Effectively Implement Requirements, Not Document Them

Too many projects are crushed by the overhead required to develop and maintain comprehensive documentation and traceability between it. Take an agile approach to documentation and keep it lean and effective. The most effective documentation is just barely good enough for the job at hand. By doing this, you can focus more of your energy on building working software, and isn't that what you're really being paid to do?

William Gens (http://www.agileconnection.com/interview/traceabilitys-priceless-role-agile-interview-william-gens)



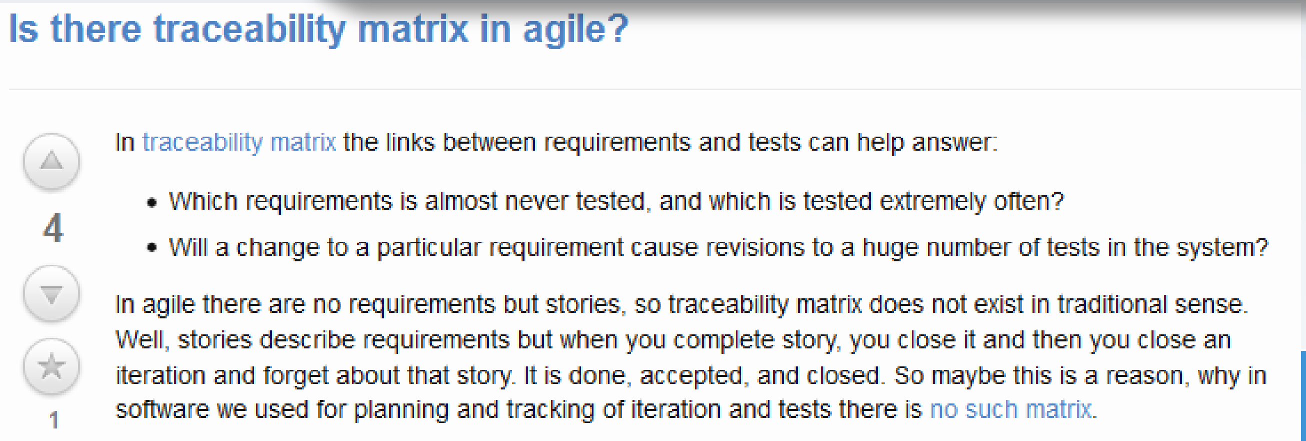
Noel: How does well executed traceability benefit project management specifically?

William: It saves time an
those artifacts are, it pro
review the requirements,
the PM with exact directi
loosely defined requirem
eavesdropping, then the
success of that specific r
have a specific footprint

In general, traceability isn't discussed by Agilistas but when it is, it is limited in scope to requirements ←→test

No guidance is given on *how* to do it nor *when* to do it

http://sqa.stackexch



Is there traceability matrix in agile?

4

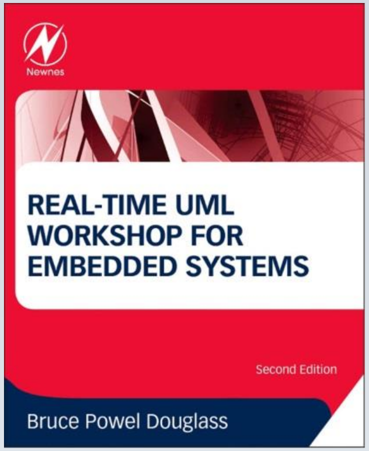
1

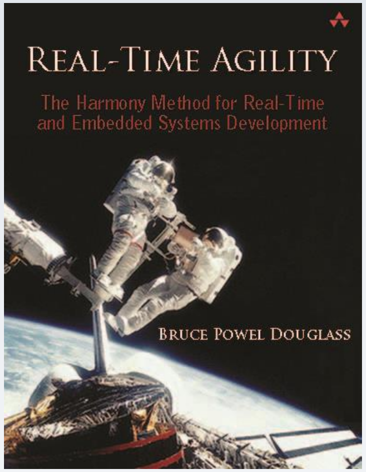
In traceability matrix the links between requirements and tests can help answer:

- Which requirements is almost never tested, and which is tested extremely often?
- Will a change to a particular requirement cause revisions to a huge number of tests in the system?

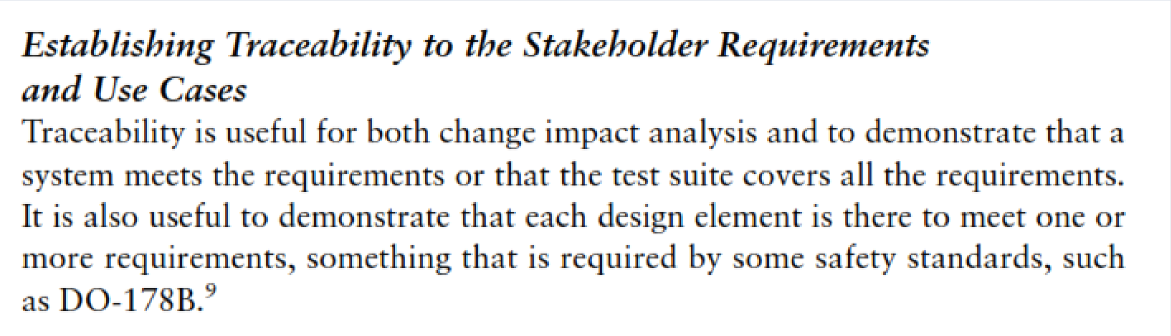
In agile there are no requirements but stories, so traceability matrix does not exist in traditional sense. Well, stories describe requirements but when you complete story, you close it and then you close an iteration and forget about that story. It is done, accepted, and closed. So maybe this is a reason, why in software we used for planning and tracking of iteration and tests there is no such matrix.

What does the Agile literature say about traceability?



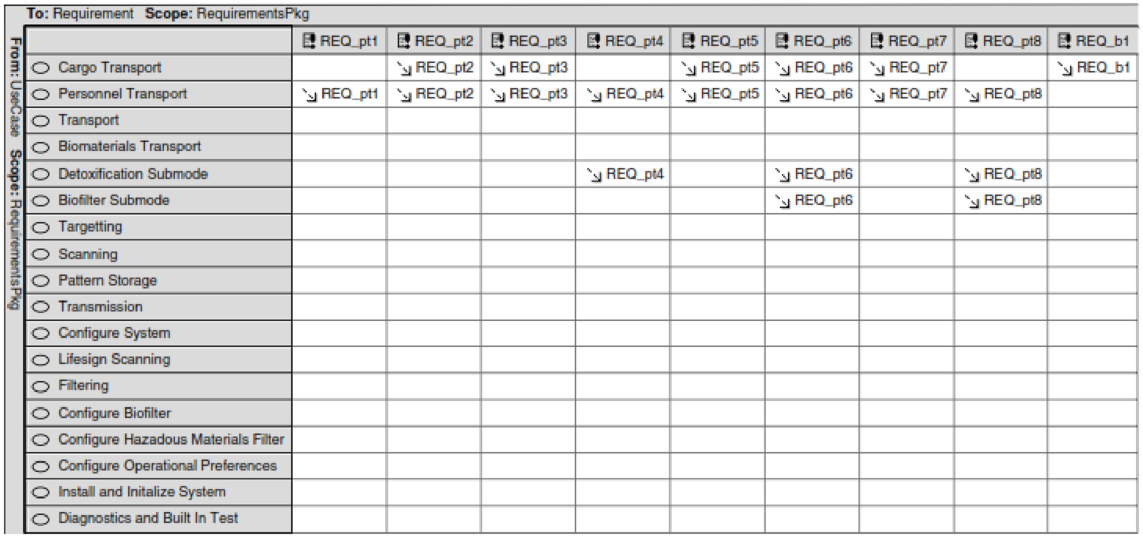

REAL-TIME AGILITY
The Harmony Method for Real-Time and Embedded Systems Development
BRUCE POWEL DOUGLASS

- Discusses traceability with respect to requirements, safety analysis, architecture, design and test



Establishing Traceability to the Stakeholder Requirements and Use Cases

Traceability is useful for both change impact analysis and to demonstrate that a system meets the requirements or that the test suite covers all the requirements. It is also useful to demonstrate that each design element is there to meet one or more requirements, something that is required by some safety standards, such as DO-178B.[9]



To: Requirement **Scope:** RequirementsPkg

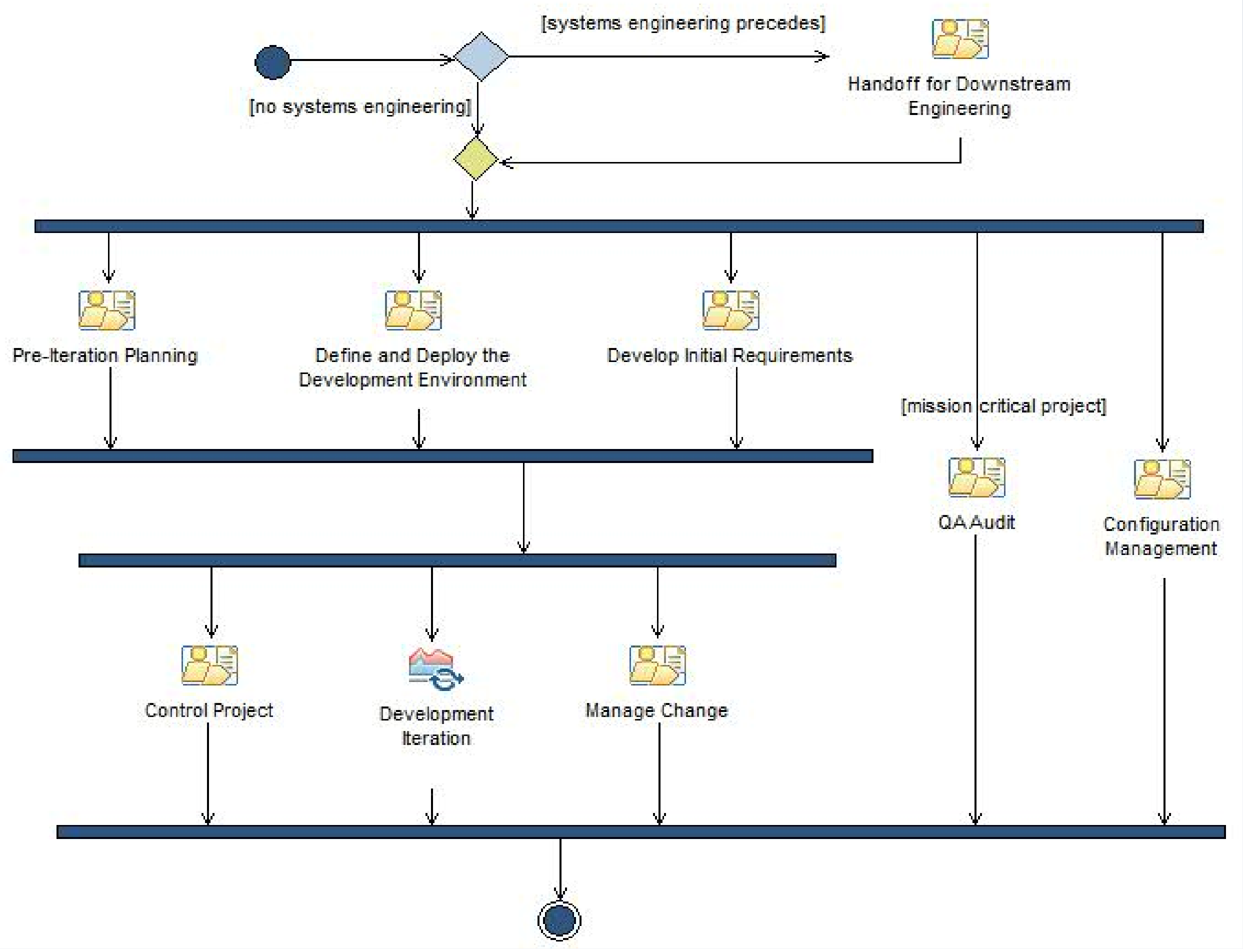
From: UseCase **Scope:** RequirementsPkg

	REQ_pt1	REQ_pt2	REQ_pt3	REQ_pt4	REQ_pt5	REQ_pt6	REQ_pt7	REQ_pt8	REQ_b1
○ Cargo Transport		↘ REQ_pt2	↘ REQ_pt3		↘ REQ_pt5	↘ REQ_pt6	↘ REQ_pt7		↘ REQ_b1
○ Personnel Transport	↘ REQ_pt1	↘ REQ_pt2	↘ REQ_pt3	↘ REQ_pt4	↘ REQ_pt5	↘ REQ_pt6	↘ REQ_pt7	↘ REQ_pt8	
○ Transport									
○ Biomaterials Transport									
○ Detoxification Submode				↘ REQ_pt4		↘ REQ_pt6		↘ REQ_pt8	
○ Biofilter Submode						↘ REQ_pt6		↘ REQ_pt8	
○ Targetting									
○ Scanning									
○ Pattern Storage									
○ Transmission									
○ Configure System									
○ Lifesign Scanning									
○ Filtering									
○ Configure Biofilter									
○ Configure Hazadous Materials Filter									
○ Configure Operational Preferences									
○ Install and Initalize System									
○ Diagnostics and Built In Test									

Figure 6.19 *Requirements Traceability Matrix*

Fitting Traceability into the Agile Lifestyle

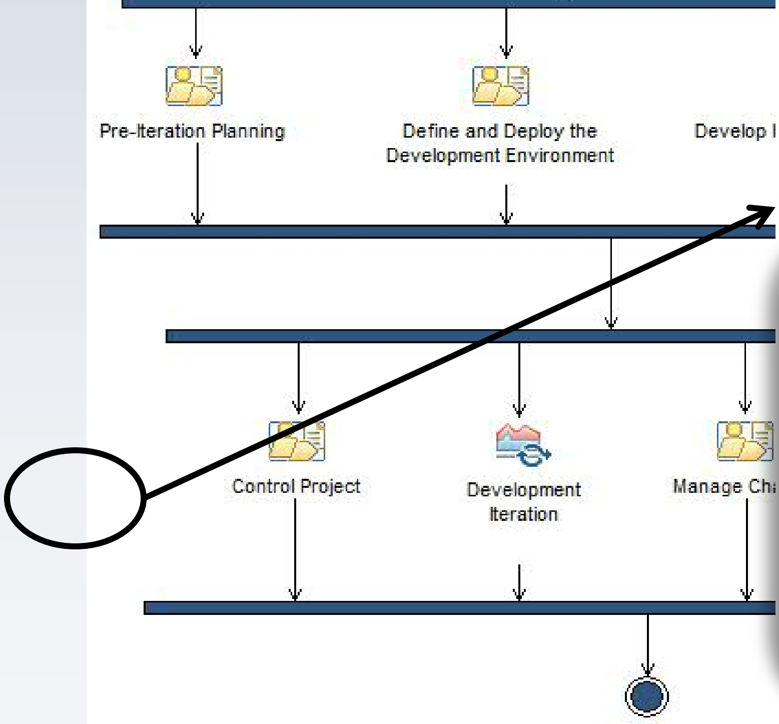
- Basic Premise: do “just enough” traceability to meet the project needs
- Best Practice: Incrementally add traceability

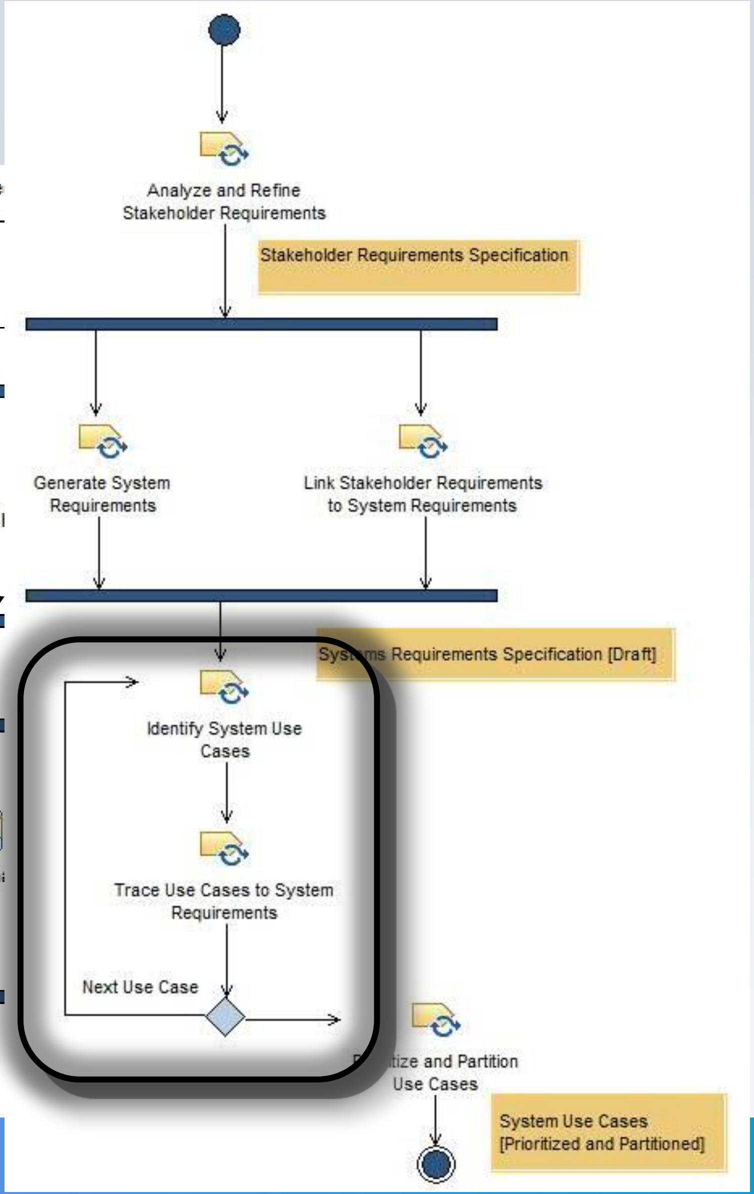

[systems engineering precedes]
Handoff for Downstream Engineering
[no systems engineering]
Pre-Iteration Planning
Define and Deploy the Development Environment
Develop Initial Requirements
[mission critical project]
QA Audit
Configuration Management
Control Project
Development Iteration
Manage Change

Fitting Traceability into the Agile Lifestyle

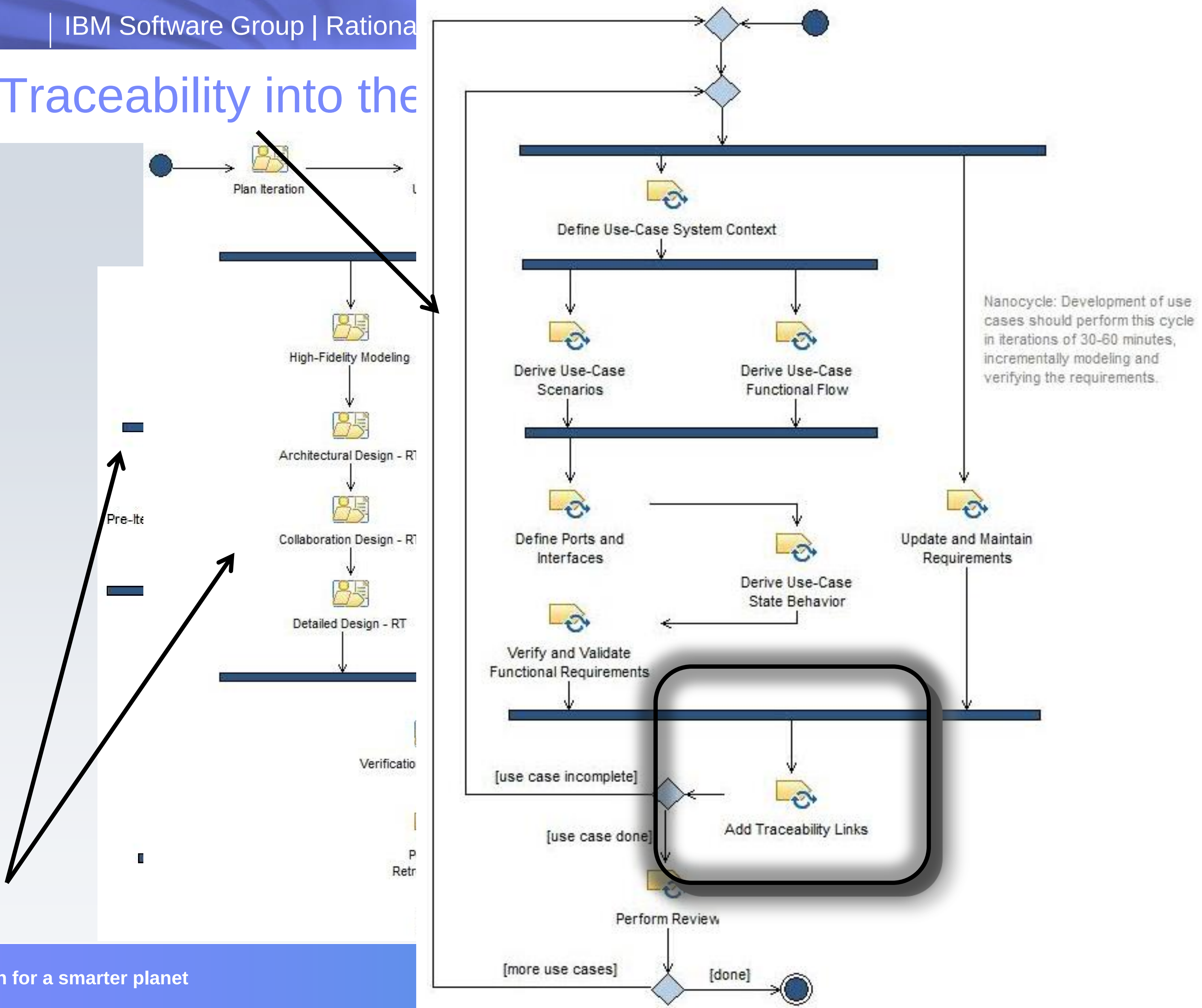
Stakeholder requirements definition

- As requirements stabilize at the use case / user story level, add traceability to relevant elements
- Iteration level is at the use case level.



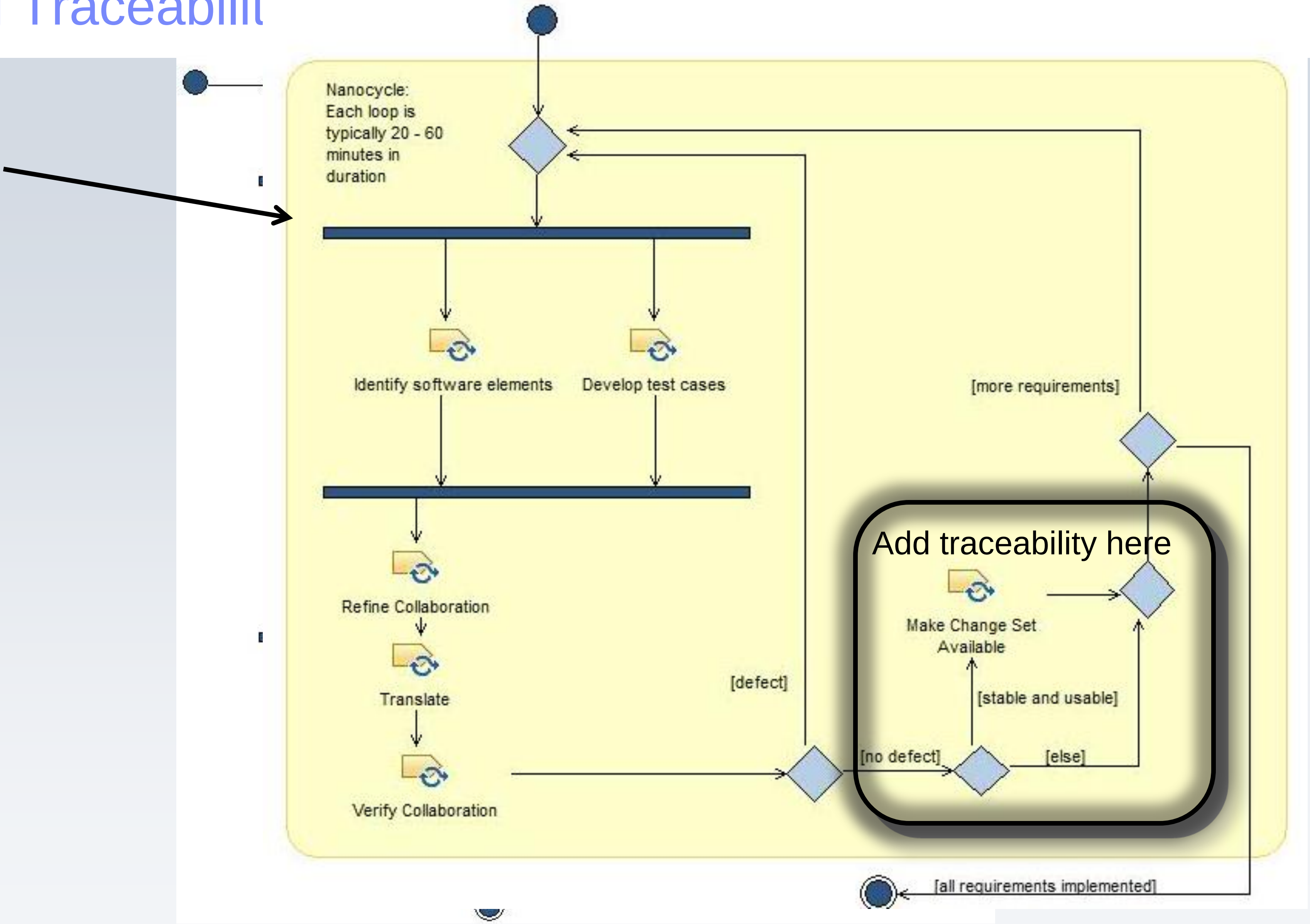

Analyze and Refine Stakeholder Requirements
Stakeholder Requirements Specification
Generate System Requirements
Link Stakeholder Requirements to System Requirements
Systems Requirements Specification [Draft]
Identify System Use Cases
Trace Use Cases to System Requirements
Next Use Case
P...tize and Partition Use Cases
System Use Cases [Prioritized and Partitioned]

Fitting Traceability into the


Plan Iteration
U
High-Fidelity Modeling
Architectural Design - RT
Collaboration Design - RT
Detailed Design - RT
Pre-Ite
Verificatio
P
Retr
Define Use-Case System Context
Derive Use-Case Scenarios
Derive Use-Case Functional Flow
Define Ports and Interfaces
Derive Use-Case State Behavior
Update and Maintain Requirements
Verify and Validate Functional Requirements
[use case incomplete]
[use case done]
Add Traceability Links
Perform Review
[more use cases]
[done]

Nanocycle: Development of use cases should perform this cycle in iterations of 30-60 minutes, incrementally modeling and verifying the requirements.

Fitting Traceabilit



Add traceability here

Traceability is essential for Agile Systems

Do it … if you want to ensure consistency among work products

… if want to do impact analysis

Provided that you add / maintain trace links incrementally

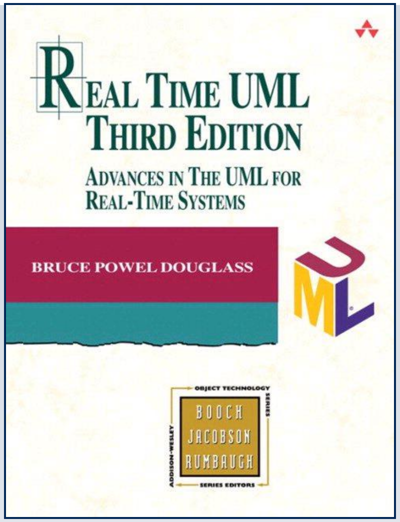
Tracing can be done manually, with RM tools like DOORS, in design tools like Rhapsody or visualized in aggregation tools such as RELM

Part of your change control process must include updating & verifying trace links

… if want to do provide evidence of compliance

… if you need to ensure testing completeness

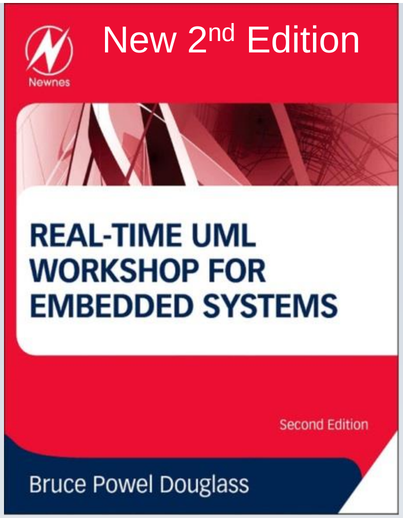
References



REAL TIME UML THIRD EDITION

ADVANCES IN THE UML FOR REAL-TIME SYSTEMS

BRUCE POWEL DOUGLASS



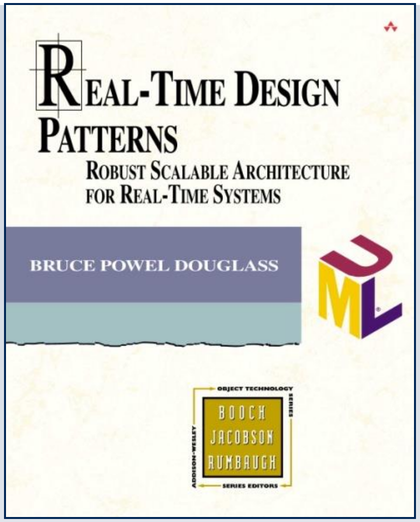
New 2nd Edition

Newnes

REAL-TIME UML WORKSHOP FOR EMBEDDED SYSTEMS

Second Edition

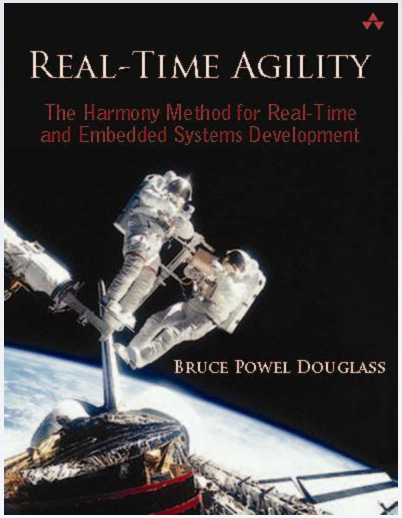
Bruce Powel Douglass



REAL-TIME DESIGN PATTERNS

ROBUST SCALABLE ARCHITECTURE FOR REAL-TIME SYSTEMS

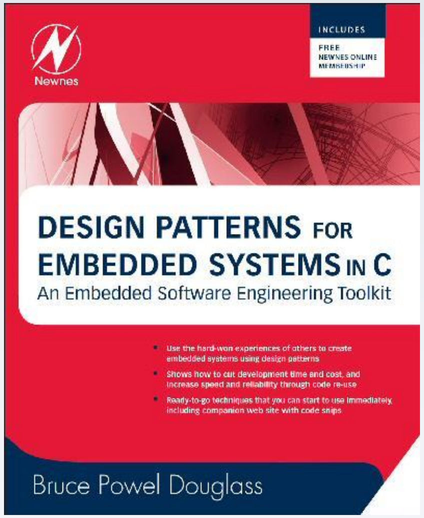
BRUCE POWEL DOUGLASS



REAL-TIME AGILITY

The Harmony Method for Real-Time and Embedded Systems Development

BRUCE POWEL DOUGLASS



Newnes

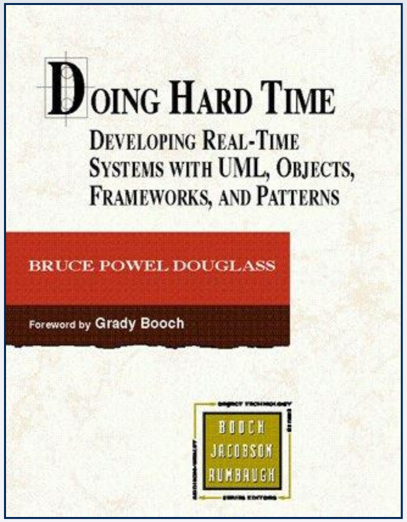
INCLUDES FREE NEWNES ONLINE MEMBERSHIP

DESIGN PATTERNS FOR EMBEDDED SYSTEMS IN C

An Embedded Software Engineering Toolkit

- Use the hard-won experiences of others to create embedded systems using design patterns
- Shows how to cut development time and cost, and increase speed and reliability through code re-use
- Ready-to-go techniques that you can start to use immediately, including companion web site with code snips

Bruce Powel Douglass



DOING HARD TIME

DEVELOPING REAL-TIME SYSTEMS WITH UML, OBJECTS, FRAMEWORKS, AND PATTERNS

BRUCE POWEL DOUGLASS

Foreword by Grady Booch