Serrie-justine Chapman (TVS)
Dr Mike Bartley (TVS)
in collaboration with
Test and Verification Solutions Ltd
Infineon Technologies UK
ARTEMIS CRYSTAL project

Requirements-driven Verification Methodology for Standards Compliance
Tutorial T6

ISO26262 (automotive) DO254 (avionics)

Requirements-driven verification
  Feature level
  Traceable
  Proven

Requirements driven vs Coverage driven
Agenda

- Requirements engineering
- Hierarchies
- Quality Gateway
- Requirements mapping
- Data Integrity
- Proof of implementation
REQUIREMENTS ENGINEERING DEFINITIONS

Requirement:
(1) A condition or capability needed by a user to solve a problem or achieve an objective

(2) A condition or capability that must be met or possessed by a system or system component to satisfy a contract, standard, specification or other formally imposed documents

(3) A documented representation of a condition or capability as in (1) or (2) [IEEE Std.610.12-1990]

Stakeholder*:
A stakeholder of a system is a person or an organization that has an (direct or indirect) influence on the requirements of the system

* All Definitions taken from IREB
Requirements Engineering:

(1) Requirements engineering is a systematic and disciplined approach to the specification and management of requirements with the following goals:

(1.1) Knowing the relevant requirements, achieving a consensus among the Stakeholders about these requirements, documenting them according to given standards, and managing them systematically

(1.2) Understanding and documenting the stakeholders’ desires and needs, then specifying and managing requirements to minimize the risk of delivering a system that does not meet the stakeholders’ desires and needs

Four core activities:

- Elicitation
- Documentation
- Validation and negotiation
- Management
REQUIREMENTS ENGINEERING
OVERVIEW

Figure: Typical Requirements Tree

- Stakeholder Requirements (Customers and internal)
- Product Requirements
- Safety Requirements
- System and Module Specs
- Verification & Test Plans
- Verification & Test Results
THE REAL REQUIREMENTS ENGINEERING OVERVIEW

Figure: Typical Requirements Tree
Agenda

- Requirements engineering
- Elicitation
- Quality Gateway
- Requirements mapping
- Data Integrity
- Proof of implementation
Where and how do we elicit the requirements, what format and quality are they?

- Identify ALL your stakeholders
- Plan and define your interface to stakeholders
- Understanding meaning/Glossary/Ontologies
- Define language/models etc
Example: A lane crossing automotive use-case

Possible Stakeholders:
- Car maker (OEM)
- ISO26262 and other standards
- Quality
- Internal users who interact with the system (CIF, ABS)
- Compliance
- Legal dept

Interface:
- Who? Application Engineers/Requirements engineers/System Architect
- How? Define a process to elicit meeting/agendas/surveys/questionnaire/brainstorming/reuse/observation etc
- What? Information is needed to enable a product – level, details

Comprehension:
- Using common standards/profiles sysML, MARTE etc
- Training needed? Safety, requirements writing
- Common glossaries

Define Languages/Models
- Excel/Models/Natural language/Formal language
- Agree Comprehension of languages/ontologies
ORIGINAL TUBE MAP
DEFINING THE SCOPE

Ensure we recognize the scope from which to elicit the requirements and any influencers outside the scope.

What are we building:
- Lane Keeping Assistant

What are we interacting with:
- GPS, CIF, ABS, Steering

What environment will we be in:
- Engine area

What dependencies and constraints do we face:
- Temperature, Voltage

What influencers do we have:
- law, environment, contract

What inputs and outputs (sources and sinks):
- GPS, CIF, warning light, actuator
FUNCTIONAL HAZARD

Function
  – The item shall optically determine the vehicles position on the road

Functional Failures
  – No Function
    • HAZARD: Doesn't stay in lane
  – Incorrect Function
    • HAZARD: Incorrectly changes lane

Situational Analysis
  – Usage situation
    • Vehicle in motion
  – People at risk
    – Vehicle occupants,
    – Pedestrians
    – cyclists
    – motorcyclists
    – occupants of other vehicles
    – motorway workers.
HAZARD ANALYSIS

Identify hazards

<table>
<thead>
<tr>
<th>Hazard</th>
<th>Doesn’t stay in lane</th>
</tr>
</thead>
<tbody>
<tr>
<td>Situation</td>
<td>unintended lane change</td>
</tr>
<tr>
<td>UID</td>
<td>123</td>
</tr>
<tr>
<td>Severity</td>
<td>S3</td>
</tr>
<tr>
<td>Rationale</td>
<td>unintended change due to speed at which the system is active or required may be life threatening to multiple parties</td>
</tr>
<tr>
<td>Exposure</td>
<td>E4</td>
</tr>
<tr>
<td>Rationale</td>
<td>Possibility of occurrence over any frequency or duration of travel in car</td>
</tr>
<tr>
<td>Control</td>
<td>C3</td>
</tr>
<tr>
<td>Rationale</td>
<td>May be required override for danger situation - short time scale to consider appropriate other actions and system not reacting to request</td>
</tr>
<tr>
<td>ASIL</td>
<td>ASIL D</td>
</tr>
</tbody>
</table>
SAFETY GOALS

Hazardous Event ID : 123

Safety goal :
  The Drivers and other road users shall not be exposed to unreasonable risk due to unintended lane change

Safety goal : Safe1

Safe State :
  The Vehicle shall remain in the lane in which they intended
HAZARD MITIGATION

Functional goal:
Undemanded Steering

ASIL level: D

Description:
The driver and other road users shall not be exposed to unreasonable risk due to un-demanded steering caused by excessive overlay torque

UID: SG001
EXTRACTING REQUIREMENTS

Functional Safety Requirement: System shall detect excessive motor torque

Definition: Excessive motor torque is defined to be the application of torque by the motor outside of the upper bounded limit of a valid torque request

Date created: 26/02/2014
UID: FSR001
Operational mode: LKA active
Allocated to elements: Controller
Fault tolerant time interval: 2 milliseconds
system fault state: limit motor torque
emergency operational interval: none
Functional redundancies: Driver override
Warning and degradation concepts: Provide a driver alert
Log a diagnostic
Deactivate system

ASIL Level: ASILD
REFINING REQUIREMENTS

• Who
  – Dependant on the organisation
  – Training
  – Safety awareness

• What Level
  – Granularity
  – Feature level for feature driven verification/test

• Review
  – Who
  – How
Agenda

- Requirements engineering
- Elicitation
- Quality Gateway
- Requirements mapping
- Data Integrity
- Proof of implementation
REQUIREMENTS AT A HIERARCHICAL LEVEL

• Where
  – Different tools
  – Different documents
  – Test benches

• How & Who
  – How – NL, Model, formal
  – Harmonisation
  – Interoperability
  – Who decides
  – How to decide

• Maintenance & Security
  – Expensive
  – Secure
  – Naming conventions
  – Change and configuration management
WASHINGTON (November 10, 1999 6:02 p.m. EST http://www.nandotimes.com) - For nine months, the Mars Climate Orbiter was speeding through space and speaking to NASA in metrics. But the engineers on the ground were replying in non-metric English.

The mathematical mismatch that was not caught until after the $125 million spacecraft, a key part of NASA's Mars exploration program, was sent crashing too low and too fast into the Martian atmosphere. The craft has not been heard from since.
REQUIREMENT QUALITY GATEWAY

- Requirements are expensive
  - ROI
  - Quality Criteria:
    - Unambiguous
    - Testable (verifiable)
    - Clear (concise, terse, simple, precise)
    - Correct
    - Understandable
    - Feasible (realistic, possible)
    - Independent
    - Atomic
    - Necessary
    - Implementation-free (abstract)

- How do we check for quality
  - Boilerplates
  - Manual inspection (review)
  - model rule checker (if model based)
Agenda

Requirements engineering

Elicitation

Quality Gateway

Requirements mapping

Data Integrity

Proof of implementation
REQUIREMENTS MAPPING

• How, what and other considerations
  – What & why
    • levels / tools /Documents
    • Essential vs non-essential
    • ROI
  – How
    • Tools
    • Review
    • Process
  – Other
    • Interfaces/Languages/protocols
    • Visibility
    • Documentation of mapping
Agenda

- Requirements engineering
- Elicitation
- Quality Gateway
- Requirements mapping
- Data Integrity
- Proof of implementation
DATA INTEGRITY

• Data management
  – Moving
  – Translating
  – Copying
  – Editing
  – Manual entry
  – Security
  – Maintenance
  – Management
DATA INTEGRITY

Requirements Database

Product Requirement Document

Internal Target Specification

Safety Concept

Test Plan

logs  checkers  coverage  test
Agenda

- Requirements engineering
- Elicitation
- Quality Gateway
- Requirements mapping
- Data Integrity
- Proof of implementation
PROOF OF IMPLEMENTATION

• Requirements stages
  – Of good quality
  – Correctly refined
  – Implemented
  – Proven to be implemented

• How to prove
  – By test
  – By review
  – By justification
  – By documentation
REQUIREMENTS DRIVEN VERIFICATION AND TEST

• Feature level Requirements – pre-requisite
• Where to store/communicate
  – Central location – access rights management
  – OSLC (Open Services for Lifecycle Collaboration)
  – ReqIF (Requirements Interface XML schema)
• Define Process/Flow..
Requirements completeness

Change management

Requirements Database

Variant x
xml

Variant x
asureSIGN

Variant x
Target Spec

Refine

Refine
REQUIREMENTS DRIVEN VERIFICATION AND TEST

• Map to tests
• Automated results analysis
  – Helps manage project
  – Helps manage data
  – Visibility
  – Data Integrity
  – Single Hierarchy
  – Closure of Requirements flow
  – Simple documentation of complete flow
  – Single solution vs complex tool landscape
DATA INTEGRITY

Requirements Engineering Flow

xml

asureSIGN

test
logs
checker
coverage

coverage

xml
## Visibility

<table>
<thead>
<tr>
<th>Domain</th>
<th>Hierarchy</th>
<th>IP</th>
<th>SOC</th>
<th>IP Val</th>
<th>SOC Val</th>
<th>Test</th>
<th>SW</th>
<th>Requirements Traceability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Variant 1</td>
<td>System</td>
<td>4 x moduleB</td>
<td><a href="mailto:soc1@tvs.com">soc1@tvs.com</a></td>
<td><a href="mailto:val1@tvs.com">val1@tvs.com</a></td>
<td>Pass</td>
<td>pass</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Variant 2</td>
<td>System</td>
<td>3 x moduleB</td>
<td><a href="mailto:soc1@tvs.com">soc1@tvs.com</a></td>
<td><a href="mailto:val1@tvs.com">val1@tvs.com</a></td>
<td>pass</td>
<td>pass</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Variant 3</td>
<td>moduleB</td>
<td>Feature A</td>
<td><a href="mailto:ip1@tvs.com">ip1@tvs.com</a></td>
<td><a href="mailto:soc2@tvs.com">soc2@tvs.com</a></td>
<td>pass</td>
<td>pass</td>
<td>pass</td>
<td>fail</td>
</tr>
<tr>
<td>Variant 3</td>
<td>moduleB</td>
<td>Feature X</td>
<td><a href="mailto:ip2@tvs.com">ip2@tvs.com</a></td>
<td><a href="mailto:iv2@tvs.com">iv2@tvs.com</a></td>
<td>pass</td>
<td>pass</td>
<td>gold</td>
<td>bronze</td>
</tr>
<tr>
<td>Variant 3</td>
<td>moduleB</td>
<td>Feature</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Asil D?
- pass
- fail

### Not tested?
- pass
- fail
Any questions?
Variant Management

Requirements Database

Variant x xml
Variant y xml
Variant z xml
Variant a xml

Import of feature level requirements
Partial import of just top-level requirements
Complete import include all mapping
Refine & map
Becomes

Copy of Variant x asureS\n\n
Copy of Variant x asureS

Variant y asureS\n\n
Variant y asureS