Requirements-driven Verification Methodology

Mike Bartley (TVS)

in collaboration with

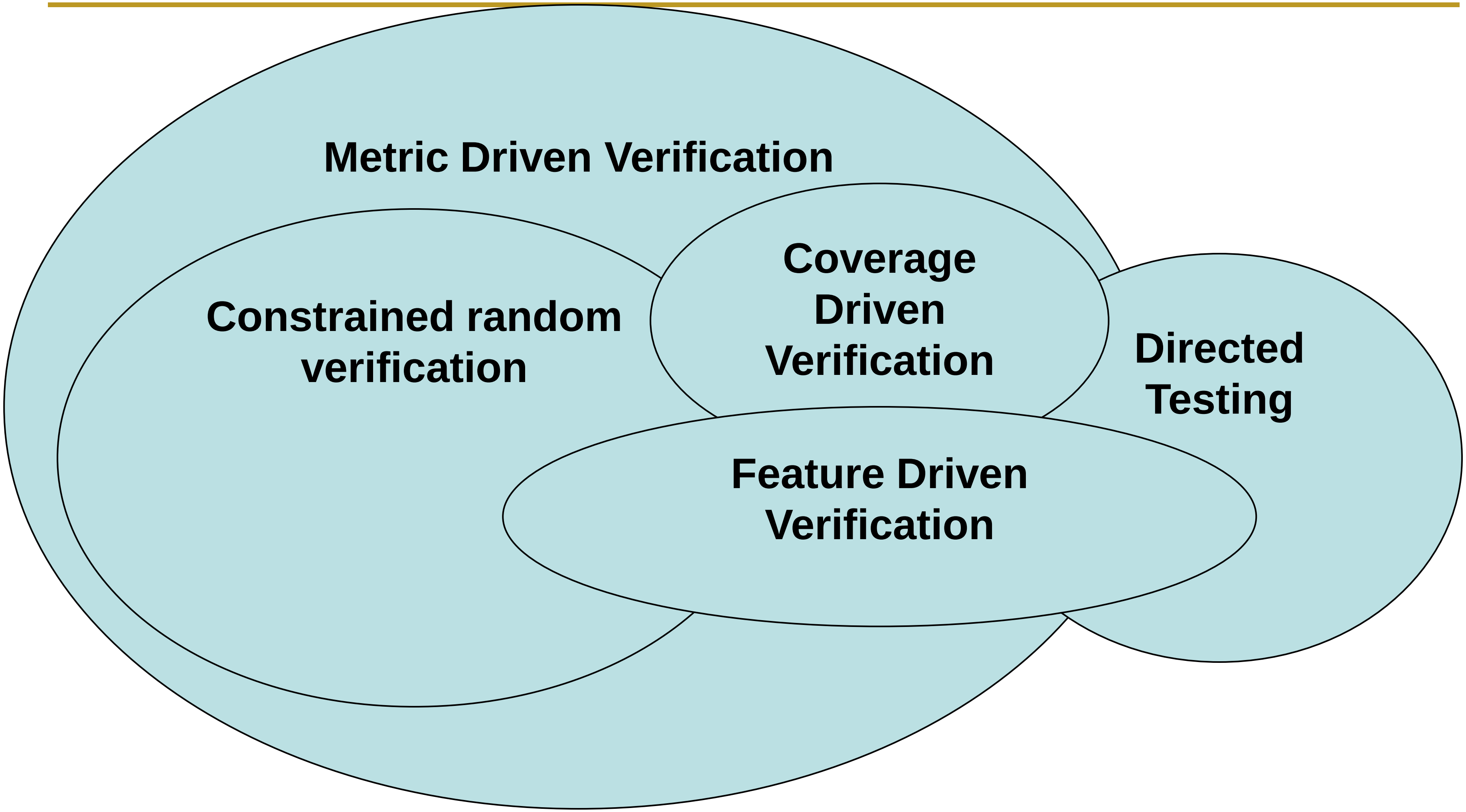
Test and Verification Solutions Ltd

ARTEMIS CRYSTAL project

Agenda

- **Motivation**
  - Why Requirements Driven Verification?
- **Introduction to Safety**
  - The Safety Standards
  - What do we need to do? And deliver?
- **Supporting Requirements Driven Verification with Advanced Verification Techniques**
- **Advantages of Requirements Driven Verification**

An Overview of Verification Approaches



- Assertion-based verification.
- Formal property based verification.

Why Requirements Driven Verification?

- **Metric Driven Verification**
  - Allows us to define targets
  - And monitor progress
- **Coverage Driven Verification**
  - Most common metric driven verification approach
  - Code Coverage
  - Functional coverage
    - Might be related to features
- **Assertion-Based Coverage**
- **These are very verification centric**
  - We measure verification progress through verification-related metrics

Why Requirements Driven Verification?

- **Feature Driven Verification**
  - Features MIGHT be related to spec
    - Is that relationship captured?
  - But are features related to requirements?

- **Requirements Driven Verification**
  - Map verification to requirements
  - Measure progress by "Requirements Working"
  - Needed for safety related domains

What is Safety?

Safety in the context of automotive embedded systems is about the prevention, detection, and response to unintended behavior that can lead to harm for the vehicle occupants and other road user

- Obvious examples:
  - anti-lock brakes, air bags, traction control, electronic cruise control, adaptive cruise control, collision avoidance, lane change control
- Less obvious examples:
  - front windshield defroster/defogger, rear windshield (backlite) defroster, auto-on headlamps, auto-on running lights, seat-belt pre-tensioners, low tire pressure warning system, engine, electric-assist power steering.

Why is Safety (and Security) important?

- **IC Insights research**
  - The automotive industry is set to drive chip demand over the coming years.
  - IC Insights research suggests the demand from automotive is expected to exhibit average annual growth of 10.8% into at least 2018.
  - Demand will come from safety features that are increasingly becoming mandatory, such as backup cameras or eCall, and the near-ubiquitous driver-assistance systems.
- **IoT**
  - Drones (avionics), autonomous cars, robots, ….
  - Connected devices have potential security threats
- **TTTech**
  - By 2020 50% of all ICs will be safety-related
  - By 2020 50% of all ICs will be connected

Safety standards

- **Industrial and Energy: IEC 61508 (2010)**
  - 61508 – core one Functional Safety of Electrical/Electronic/Programmable Electronic Safety-related Systems
- **Nuclear: IEC 61513 (2010)**
  - 61513 Nuclear power plants. Instrumentation and control important to safety. General requirements for systems
- **Avionics: DO-254 (2005) & DO-178C (2012)**
  - 254 Design assurance guidelines for airborne Electronic Hardware
  - 178C Software Considerations in Airborne Systems and Equipment Certification
- **Rail: EN 50128 (2011)**
  - Software for railway control and protection systems
- **Medical: IEC 60601-1-11 (2010)**
  - for the safety and effectiveness of medical electrical equipment 50128
- **Automotive: ISO 26262 (2011 [2018])**
  - Functional Safety standard – next one incl. motorbikes and 3.5 + tonnes

Safety Functions and Integrity

Functional safety is achieved by:

- **Safety function requirements**
  - From Hazard Analysis
- **Safety integrity requirements**
  - From Risk Assessment
- **Any system that carries out safety functions is a *Safety-related System***
- **May be separate from a control system or control system itself may be a safety-related system**
- **Higher levels of safety integrity => greater rigour in developing a system**

Safety Integrity

- **Risk based approach**
- **Depends on:**
  - severity of injur(ies)
  - frequency/duration of exposure to hazard
  - controllability of hazardous event by driver or other traffic participant
- **Degree of certainty necessary that safety function(s) will be carried out**

ISO 26262 Risk Graph

		C1	C2	C3
S1	E1	QM	QM	QM
	E2	QM	QM	QM
	E3	QM	QM	ASIL A
	E4	QM	ASIL A	ASIL B
S2	E1	QM	QM	QM
	E2	QM	QM	ASIL A
	E3	QM	ASIL A	ASIL B
	E4	ASIL A	ASIL B	ASIL C
S3	E1	QM	QM	ASIL A
	E2	QM	ASIL A	ASIL B
	E3	ASIL A	ASIL B	ASIL C
	E4	ASIL B	ASIL C	ASIL D

Severity
Exposure (probability)
Controllability

TVS

Sample Differences in Safety Integrity Levels

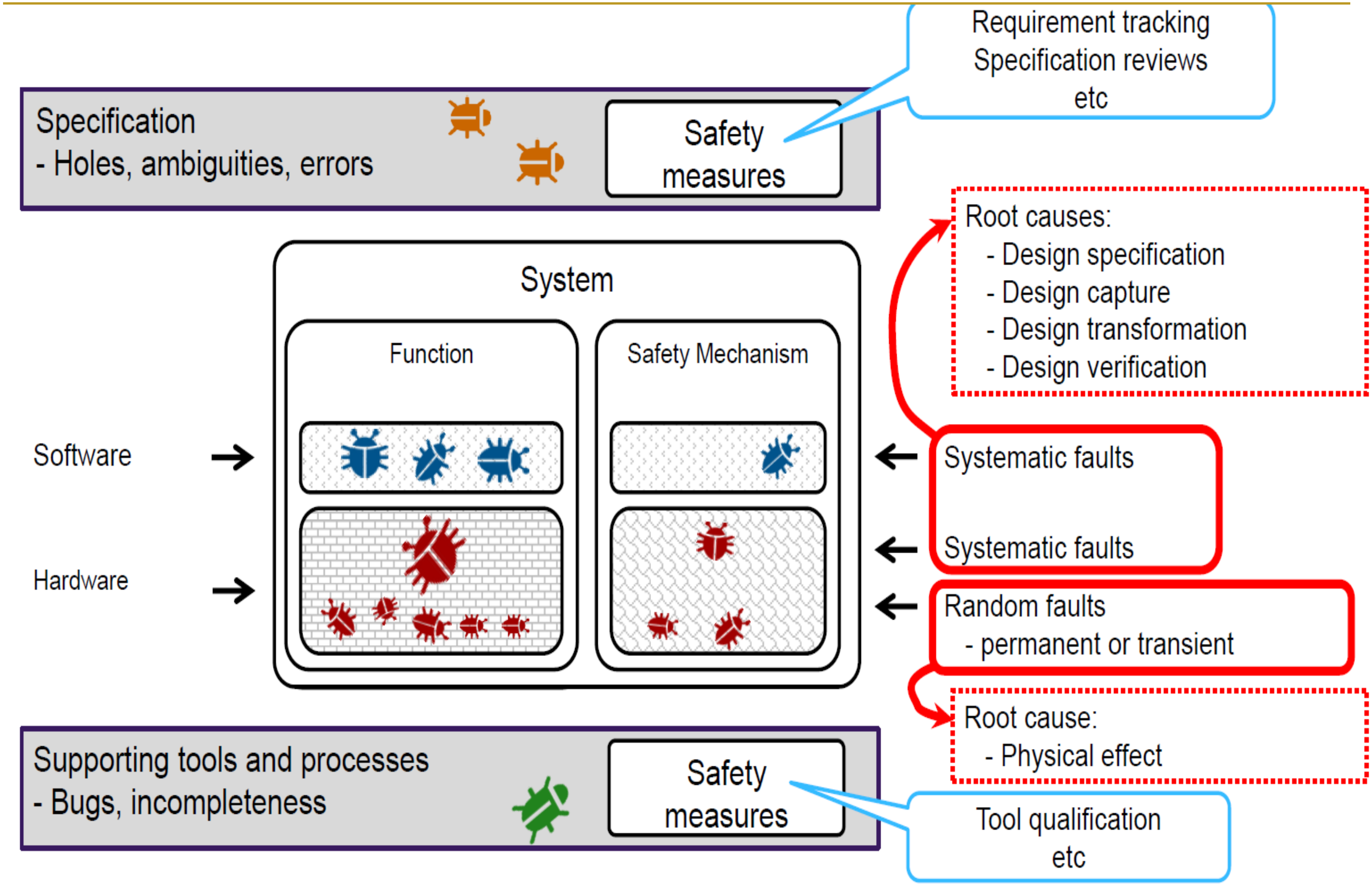
- **Dynamic analysis and testing**

Technique	SIL 1	SIL 2	SIL 3	SIL 4
Structural test coverage (entry points) 100%	HR	HR	HR	HR
Structural test coverage (statements) 100%	R	HR	HR	HR
Structural test coverage (branches) 100%	R	R	HR	HR
Structural test coverage (conditions, MC/DC) 100%	R	R	R	HR
Test case execution from boundary value analysis	R	HR	HR	HR
Test case execution from error guessing	R	R	R	R
Test case execution from error seeding	-	R	R	R
Test case execution from model-based test case generation	R	R	HR	HR
Performance modelling	R	R	R	HR
Equivalence classes and input partition testing	R	R	R	HR

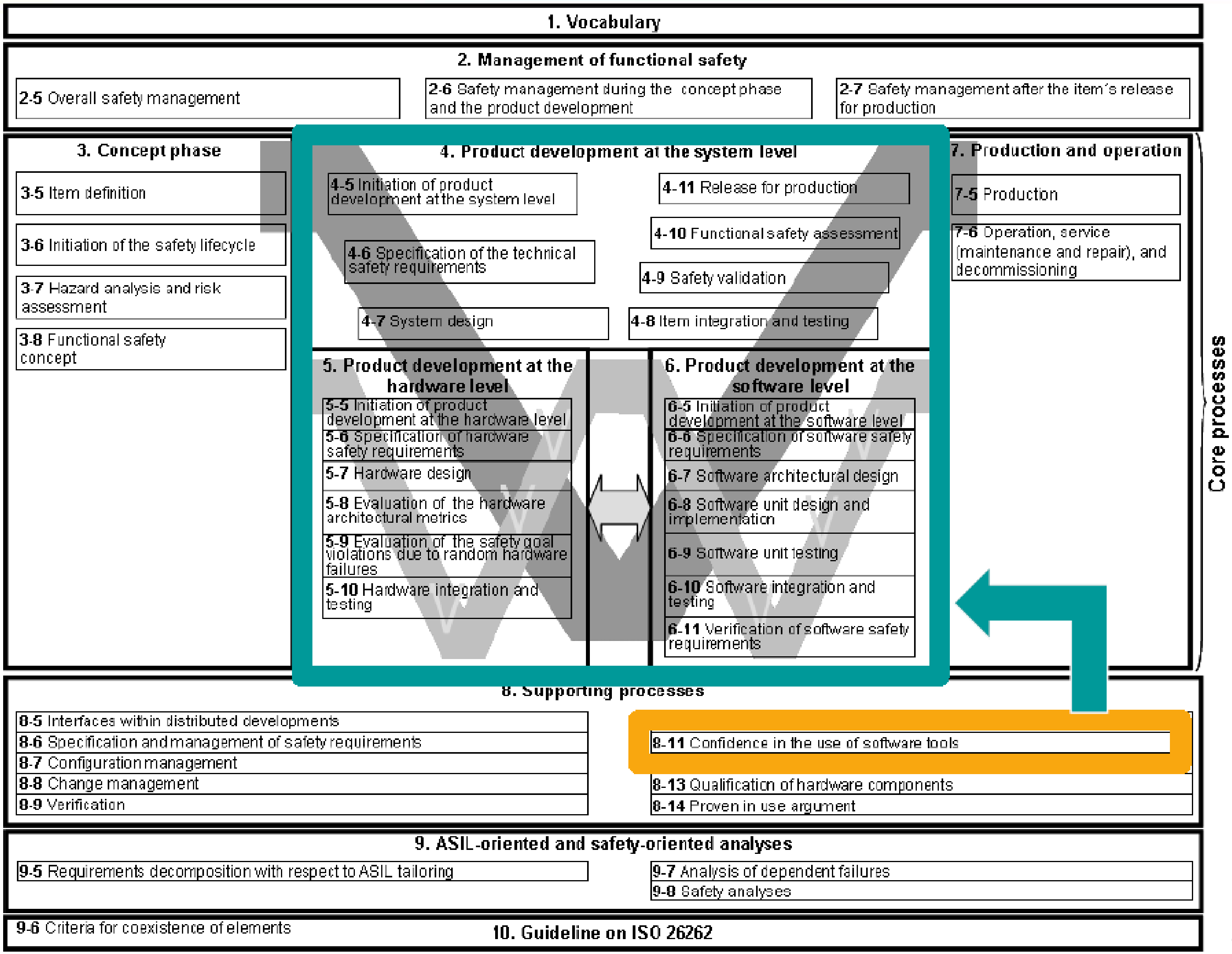
How Systems Fail

- **Random failures**
  - Can usually predict (statistically)
  - Can undertake preventative activities
- **Systematic failures**
  - Specified, designed or implemented incorrectly
  - Can’t usually predict
- **Systemic failures**
  - Shortcomings in culture or practices

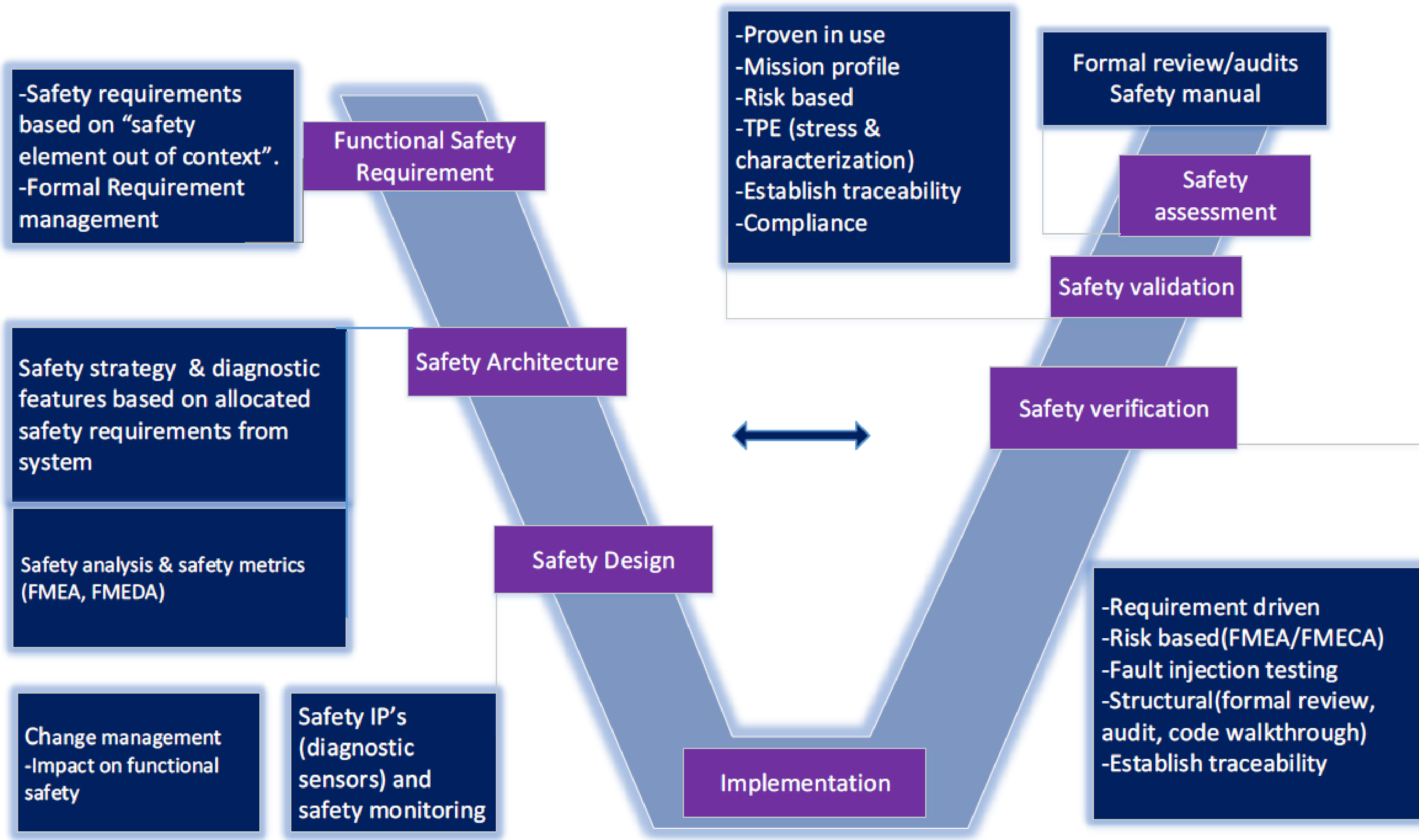
How Systems Fail



ISO2626 Overview



V Model with Safety Extension



Key Processes

- **Plans & Standards**
- **Requirements**
- **Design Specifications**
- **Reviews and Analyses**
- **Testing (against specifications)**
  - At different levels of hierarchy
- **Test Coverage Criteria**
- **Requirements Traceability**
- **Independence**

Key Deliverables

- Verification Plan
- Validation and Verification Standards
- Traceability Data
- Review and Analysis Procedures
- Review and Analysis Results
- Test Procedures
- Test Results
- Acceptance Test Criteria
- Problem Reports
- Configuration Management Records
- Process Assurance Records

Defined, traceable and controlled process

- Management
  - Safety management, ensuring culture and adherence, defining roles and responsibilities, distributing development and documenting reviewing etc
- Engineering
  - ensuring good safety design such as freedom from interference, fault injection, ecc error correction/detection etc
- Development interfaces
  - Hardware to software, pre-silicon to post, IP to SOC etc .. May differ in process and include interface process
- Verification
  - may use common methodologies such as UVM or AGILE etc – documented and proven
- Validation
  - Evidence of compliance with Safety goals and that they are correct
- Functional safety audit
  - evaluates the implementation of the processes required for the functional safety
- Functional safety assessment
  - evaluates the functional safety achieved by the item.

Stage one: What

Requirements Management

- "The management of safety requirements includes managing requirements, obtaining agreement on the requirements, obtaining commitments from those implementing the requirements, and maintaining traceability."

Table 2 — Notations for software architectural design

	Methods	ASIL			
		A	B	C	D
1a	Informal notations	++	++	+	+
1b	Semi-formal notations	+	++	++	++
1c	Formal notations	+	+	+	+

- 1.117 semi-formal notation
  - description technique whose syntax is completely defined but whose semantics definition can be incomplete
  - EXAMPLE System Analysis and Design Techniques (SADT); Unified Modeling Language (UML).

Stage two: How will we prove it

Verification

- "**6.4.3.3** An appropriate combination of the verification methods listed in Table 2 shall be applied to verify that the safety requirements comply with the requirements in this clause and that they comply with the specific requirements on the verification of safety requirements within the respective parts of ISO 26262 where safety requirements are derived."

Table 2 — Methods for the verification of safety requirements

Methods		ASIL			
		A	B	C	D
1a	Verification by walk-through	++	+	o	O
1b	Verification by inspection	+	++	++	++
1c	Semi-formal verification[a]	+	+	++	++
1d	Formal verification	o	+	+	+

[a] Method 1c can be supported by executable models.

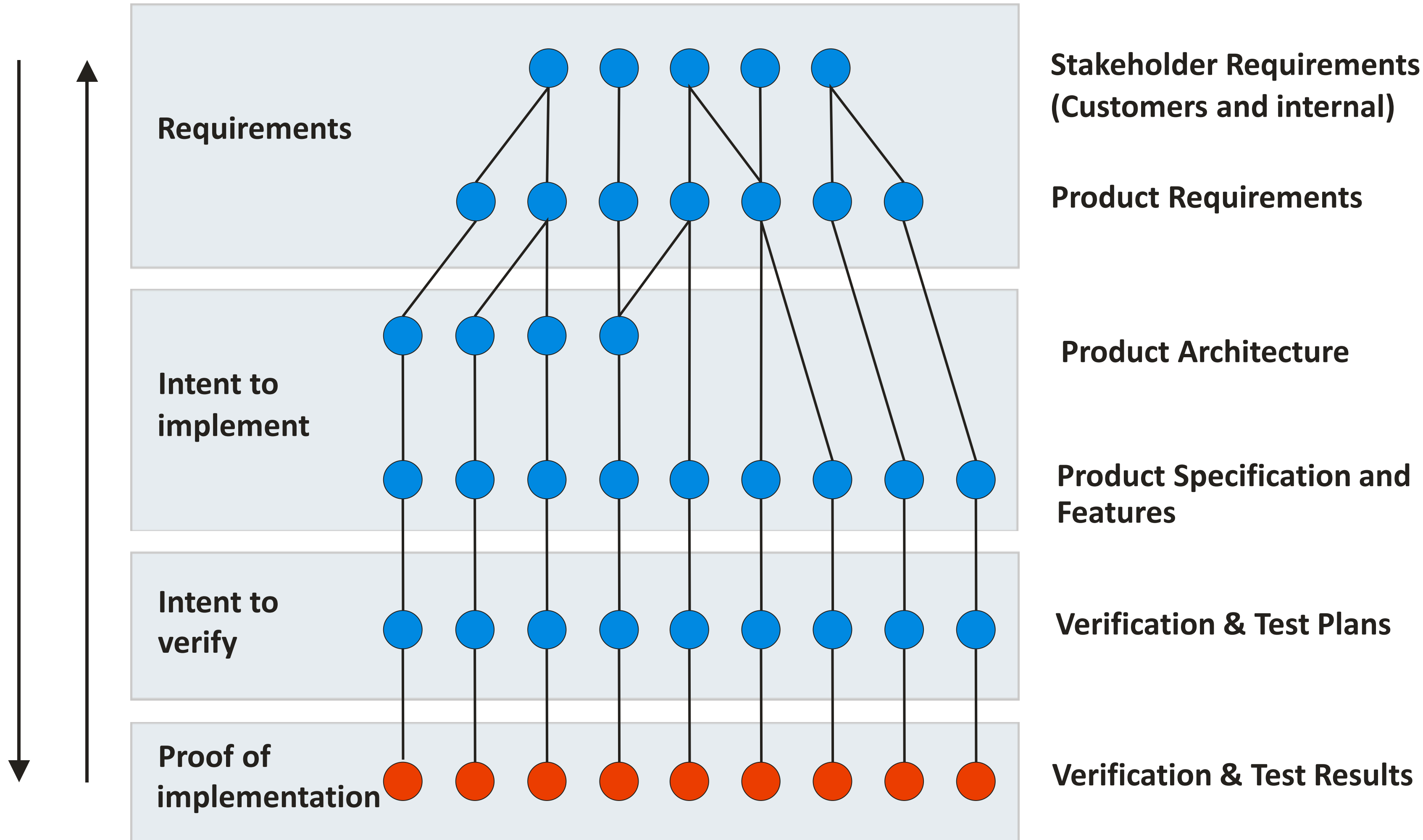
Stage Three: Proof of Implementation

- Requirements stages
  - Of good quality
  - Correctly refined
  - Implemented
  - Proven to be implemented
- How to prove
  - By test
  - By review
  - By justification
  - By documentation

Retention of Verification Results (DO 254)

- **Many objectives related to verification can be accomplished through reviews, analysis or test**
  - The results of these activities must be retained.
  - Review of the final verification results (tests and analysis) is an area that applicants commonly miss or do not give proper emphasis.
- **Verification records should contain a clear correlation to the pass/fail criteria**
  - These verification records should contain the author/reviewer, date, and any items used in the including their versions.
  - Any failures or issues found should be correlated to the standard that has been violated.
- **Test results should be clearly linked to their associated tests and requirements**
- **Test Results should be reviewed to be sure that the actual and expect results are giving the correct results and that the tests are passing.**

Traceability in Practice



Shows a mapping from features to verification and test plans

Requirements Traceability Matrix

Requirements Traceability Matrix

Requirements Traceability Matrix			Root Folder: Contract processing	**Requirement**	Agree on	Check	Create contact	Determine	See customer off	Send contact	Sign contact	Determine net price	Inform customer	Send original	Contract processing	Check if	Develop proposal	Explain contact	Quotation	Sales order
				#	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
		Total	Req		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Root Folder: Modeling			Covered		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Test	#	Test		Relate	4	4	4	4	4	4	4	2	2	2	1	0	0	0	0	0
Contact processing - path 2	1	1	X	10	X	X	X					X	X							
Contact processing - path 1	2	1	X	8				X	X	X	X									
Agree on	3	1	X	2	X															
Check	4	1	X	2		X														
Create contact	5	1	X	2			X													
Determine	6	1	X	2				X												
See customer off	7	1	X	2					X											
Send contact	8	1	X	2						X										
Send original	9	1	X	2										X						
Sign contact	10	1	X	2							X									
Contact processing	11	1	X	1											X					

- **Tables like this are popular**
  - Excel is often used
- **But it is hard to capture all the information**
  - complex relationships
  - history

The V&V Challenge

- **Cyber Physical Systems introduce a complex software testing challenge**
  - A large input space
  - Difficulty predicting expected response
- **Hardware faced a similar problem 20 years ago**
  - Over the past 20 years a number of "Advanced Hardware Verification Techniques" (AHVT) have been introduced
  - To automate test generation and response checking
- **Can this be done within a safety framework?**

The Innovate UK Research Project

- **Investigate the feasibility of applying Advanced Hardware Verification Techniques to the testing of software for Cyber Physical Systems**
  - Technical feasibility
  - Market feasibility
- **TVS**
  - Producing tools for evaluation by end user partners

University of BRISTOL

Test generation from formal models

dyson

Robotic Vacuum Cleaner

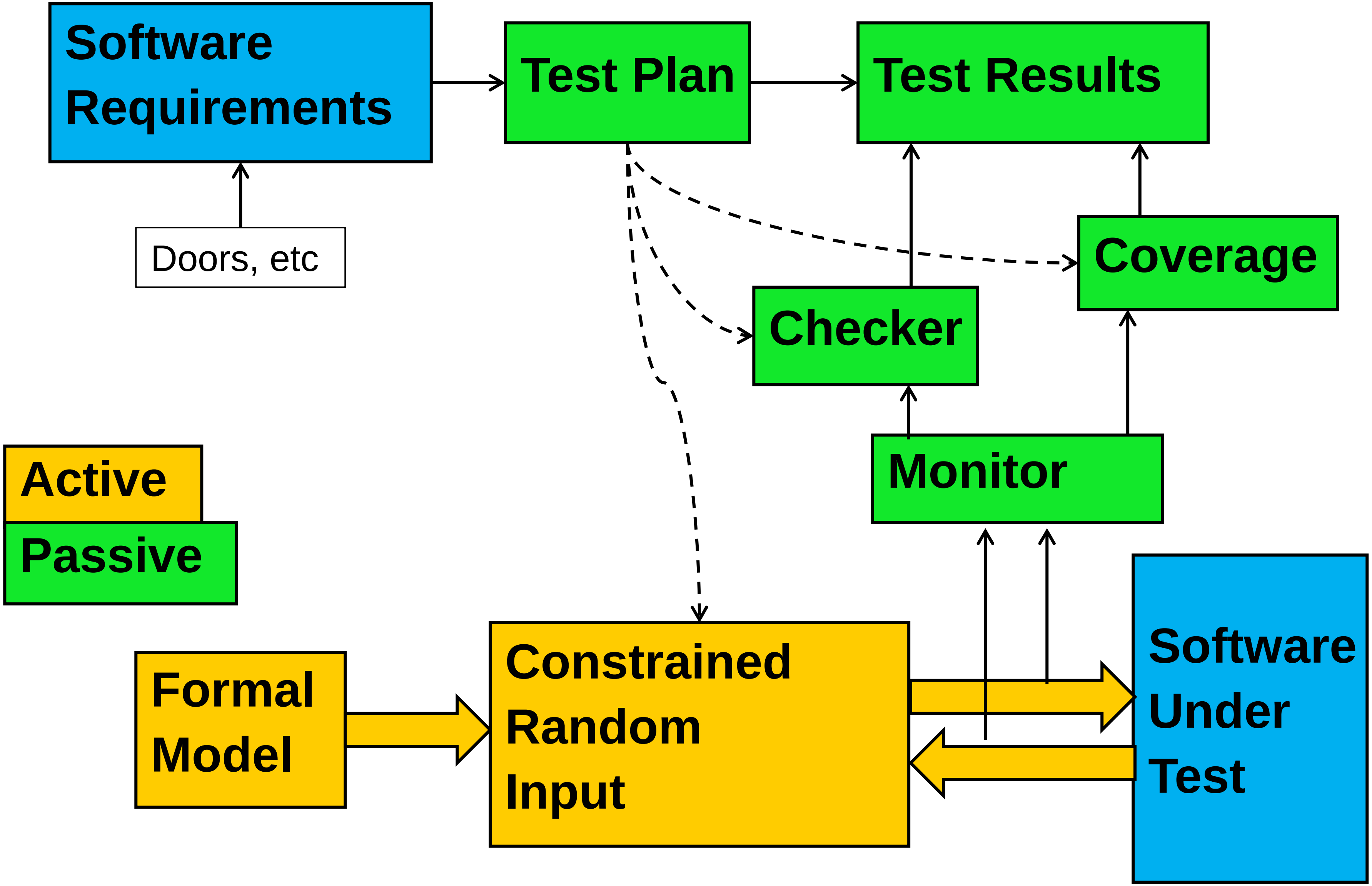
SCISYS

Software for Autonomous Vehicles

THALES

Autonomy and Offboard Systems

Advanced Hardware Verification Techniques



Results of Bubble Sort "Proof of Concept"

- Check output list is ordered
- Output list contents == input list contents

Lists of
- Integers
- Floats
- Ascii
- etc

Checkers

List Generator

Software Under Test

Lists

Constrain towards
- Empty lists
- Equal values
- Reverse ordering

Coverage Metrics

- Empty List
- Reverse ordered
- Error cases (mix integers, floats, ascii
- etc

Example Constrained Random Inputs

- **Mimic sensor input data**
- **Need to constrain those inputs**
  - Only the legal space
  - Hit the corner cases
- **Example scenarios**
  - Valid ranges for data
  - Relationships between inputs
  - Next input within certain “distance” to prior input

Functional Coverage

From Kerstin Eder of the University of Bristol

- **Requirements coverage**
- **“Cross-product” coverage**

  [O Lachish, E Marcus, S Ur and A Ziv. Hole Analysis for Functional Coverage Data. Design Automation Conference (DAC), June 10-14, 2002, New Orleans, Louisiana, USA.]

  A cross-product coverage model is composed of the following parts:

  1. A semantic **description** of the model (story)
  2. A list of the **attributes** mentioned in the story
  3. A set of all the **possible values** for each attribute (the attribute value **domains**)
  4. A list of **restrictions** on the legal combinations in the cross-product of attribute values

  A **functional coverage space** is defined as the Cartesian product over the attribute value domains.

- **Situation coverage**

  [R Alexander et al. Situation coverage – a coverage criterion for testing autonomous robots. University of York, 2015]

	┬	┤	┘	┐	─	╹	╴	┼	┴	├	└	┌	│	╻	╶
Car															
Bike															
HGV															
Ped															

Example Checkers

- **Do not accelerate too fast**
  - Assert that output to motor is not too high
- **“always respond correctly”**
  - If A&B&C occur then check X happens
    - Assertion coverage “check A&B&C occurs” for free
- **Always safe**
  - Do not get too close to other objects
  - Requires some level of modelling
- **Minimise resources**

Extracting Requirements Example

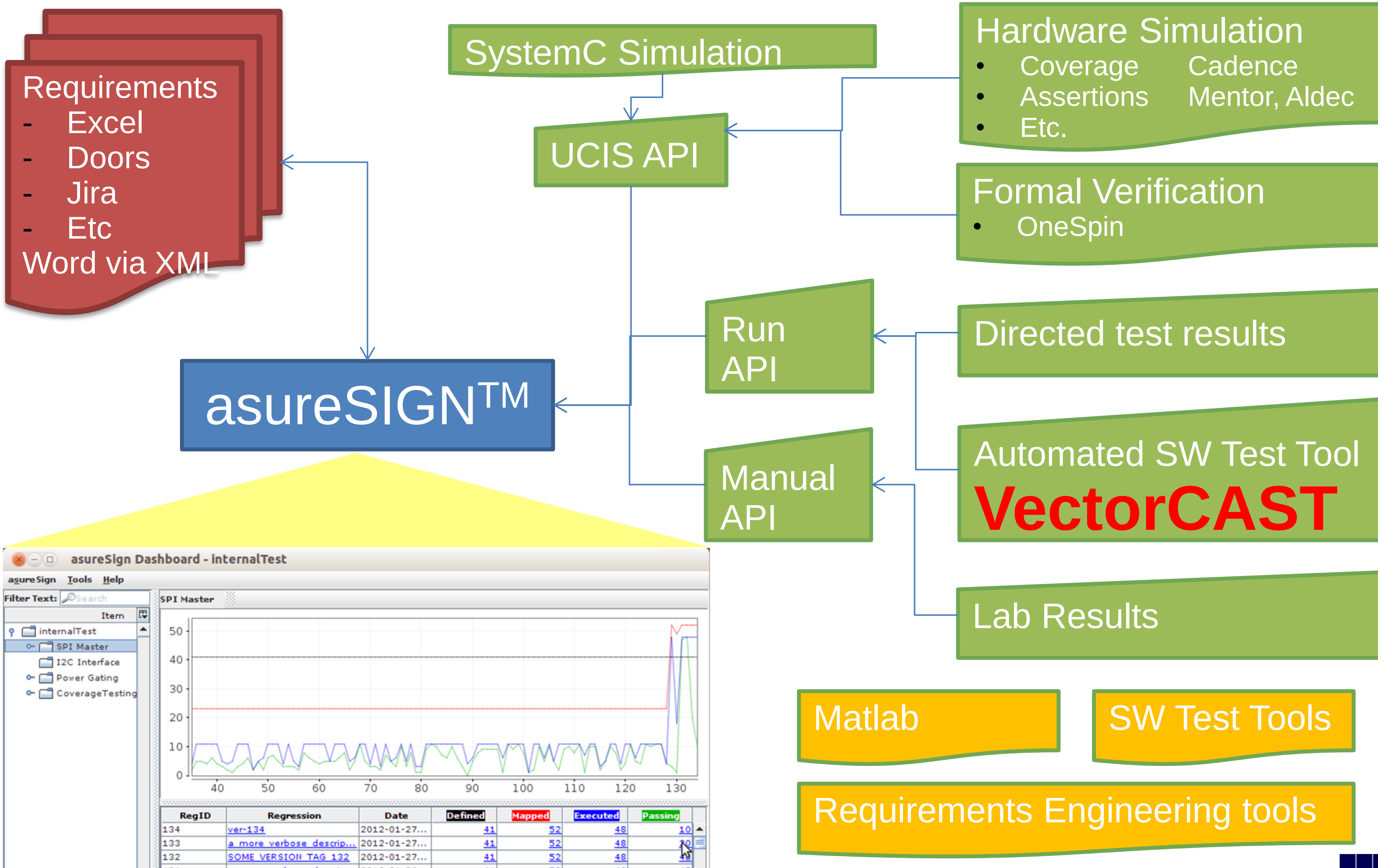
Field	Data
Functional Safety Requirement	System shall manage excessive motor torque
Feature 1	Provide a driver alert
Sub-feature 1.1	Detect excessive torque
Verification Goals	• Cover points on inputs from torque sensor • Assertion – if torque sensor input above a certain threshold then generate input • Property – prove above assertion • Software – detect the interrupt and call handler • Coverage – hardware and software MC/DC
ASIL Level	ASILD

TVS

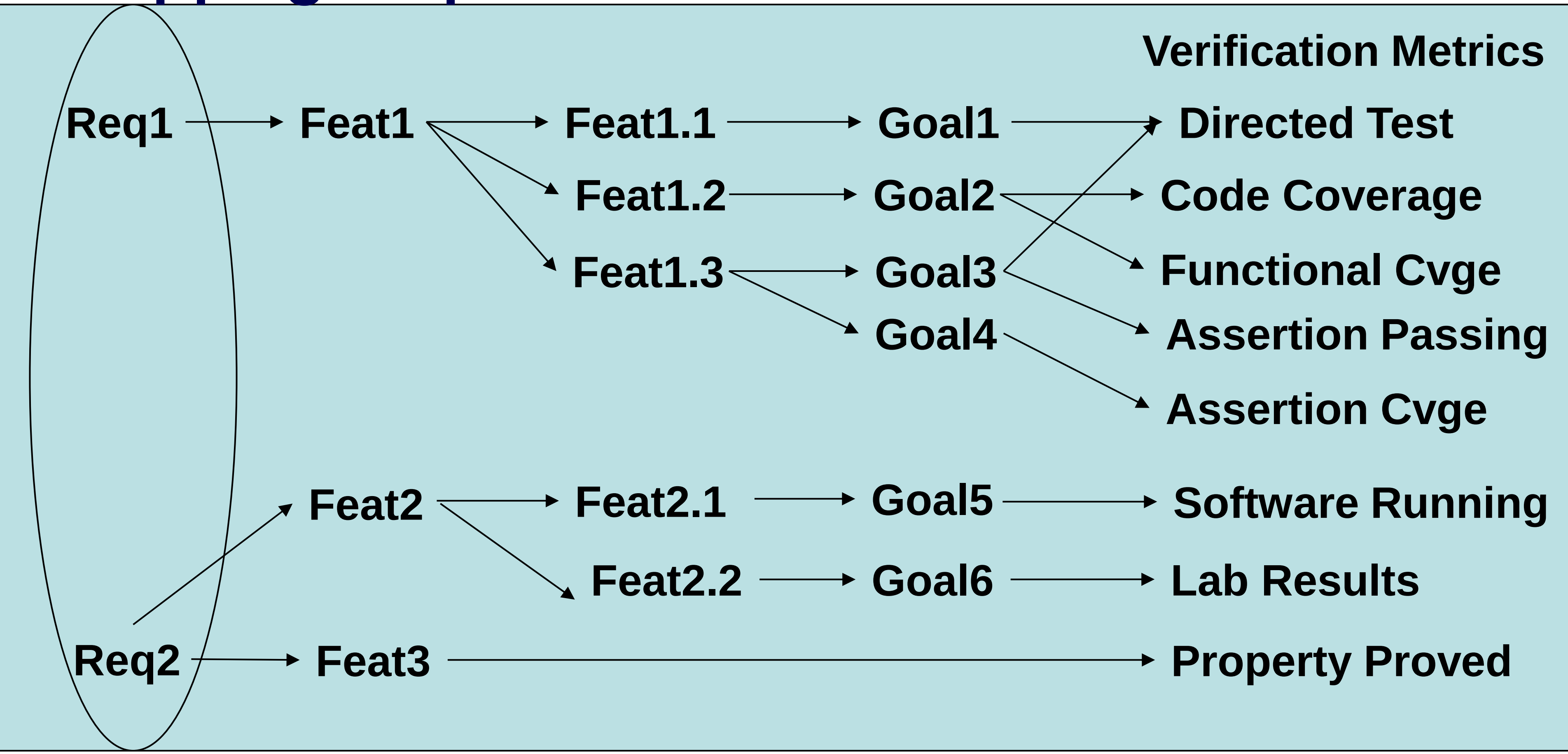
Safety compliance (asureSIGN)

- **Managing Requirements**
  - Importing and editing requirements
- **Decomposing requirements to verification goals**
- **Tracking verification execution**
  - Automating import of verification results (VectorCAST)
  - Automate accumulation and aggregation of verification results
- **Impact analysis**
  - Managing changes in requirements and verification
- **Demonstrating safety compliance – for example**
  - DO254/178C, ISO26262, IEC 60601, IEC 61508, EN 50128, IEC 61513
- **Supply chain management**
  - Exporting requirements and test plans
  - Importing test results

asureSIGN™ at the heart of HW/SW V&V



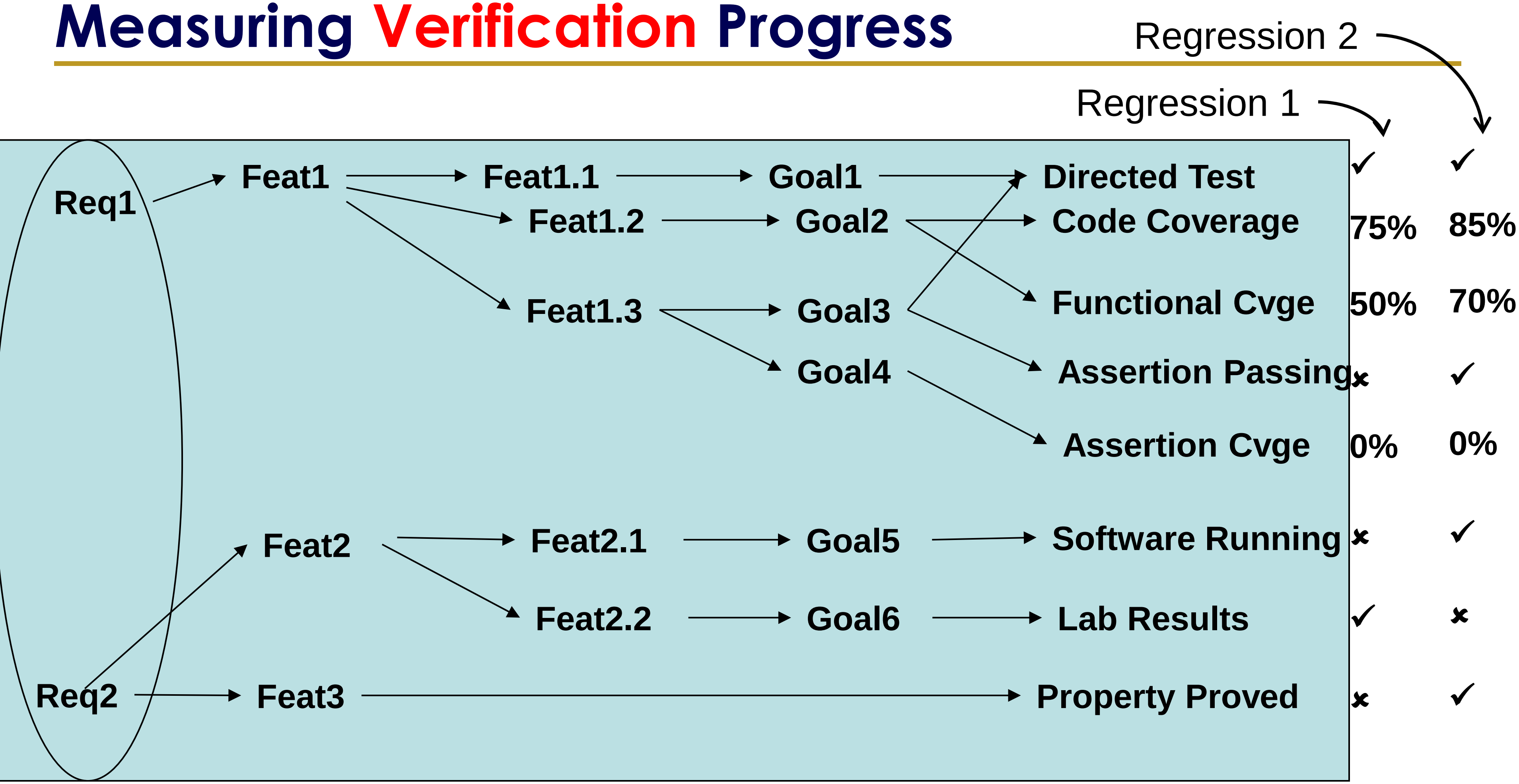
Mapping Requirements to Verification Metrics



Metrics can be:

- **From HW verification**
- **From Silicon validation**
- **From SW testing**

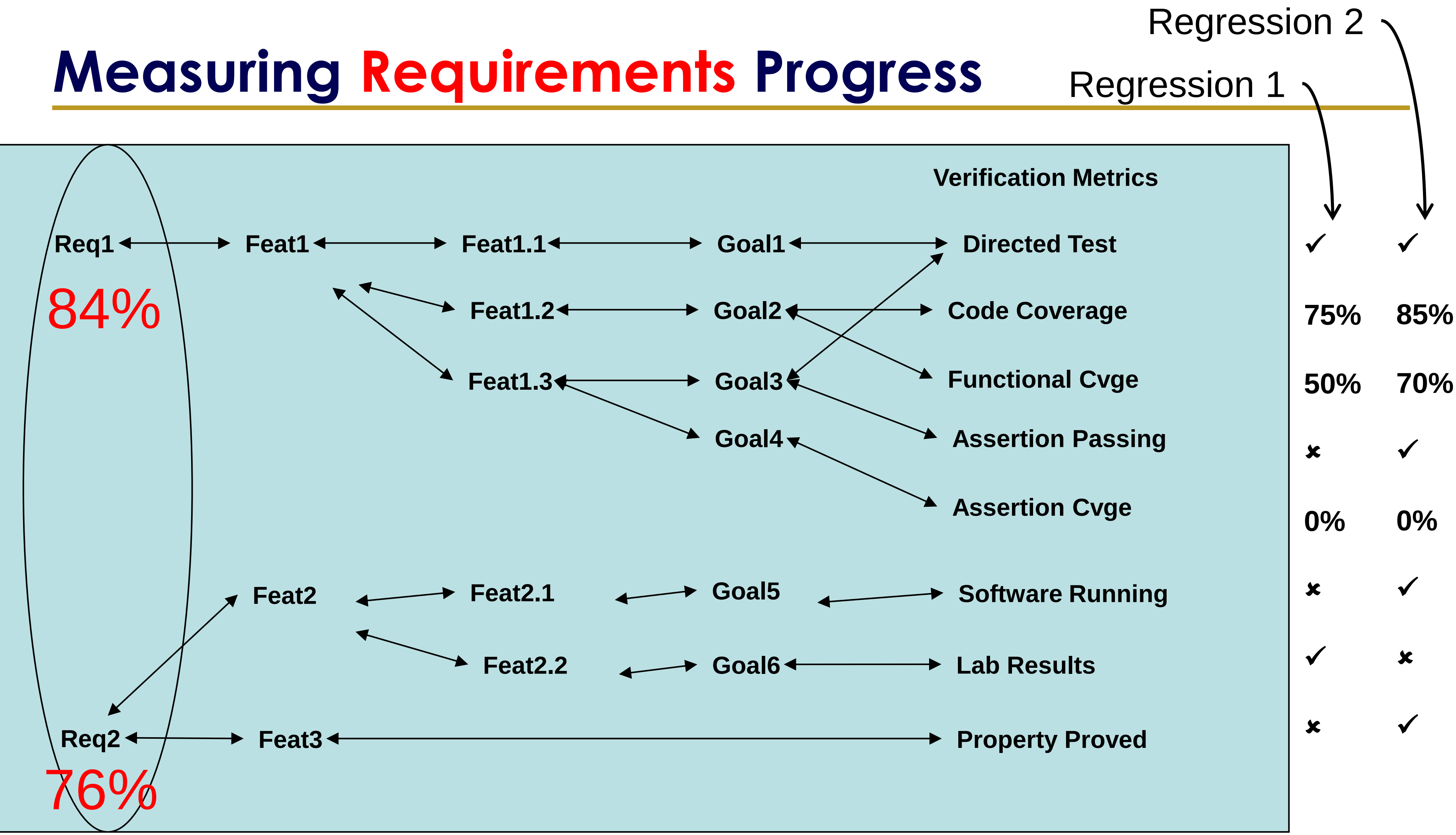
Measuring Verification Progress



Want to

- **Capture the metrics associated with verification tasks**
- **Capture progress**

Measuring Requirements Progress



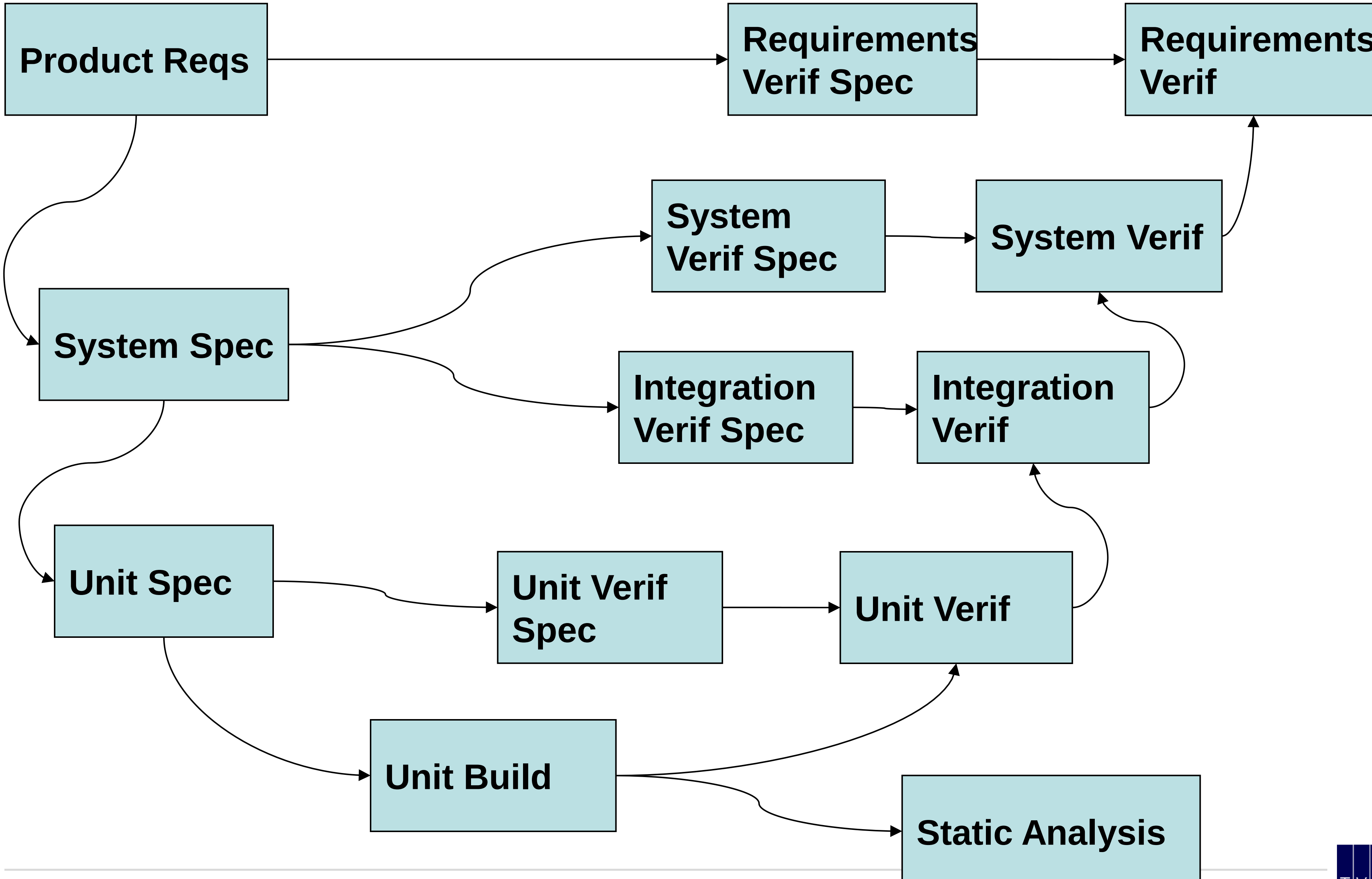
Use a bi-directional mapping to track backwards

Use an SQL database to hold the mappings and results

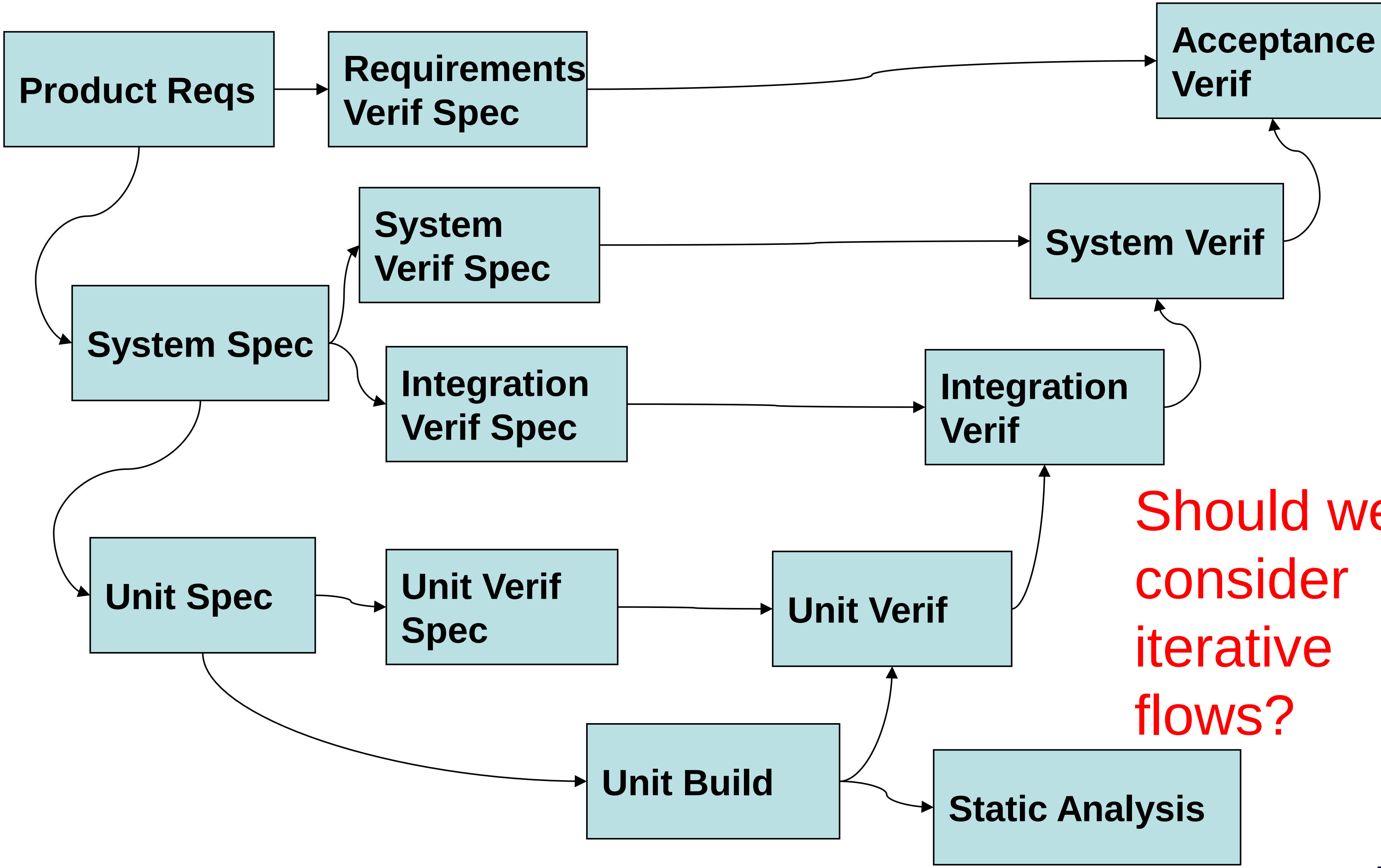
Advantages of Requirements Driven Verif

- **Identify test holes and test orphans**
- **Track the status of the whole verification effort**
  - planning, mapping, writing, execution
- **Better reporting of requirements status**
- **Requirements Prioritisation**
- **Risk-based testing**
- **Filtering Requirements based on**
  - Customers
  - Releases
- **Impact analysis**
- **Build historical perspective for more accurate predictions**

Sequential Development Flow



Shift-Left "Sequential" Development Flow



Should we consider iterative flows?

Summary

- **Requirements Driven Verification**
- **Required for Safety Related Domains**
  - Avionics, Automotive, Nuclear, Rail, Industrial
- **Advantages**
  - Report Requirements process rather than verification metrics
  - Under and over engineering
  - Shift-left for TTM improvement
- **There are a number of challenges to overcome**
  - Consider your tooling requirements
  - And retention of verification results

Any questions ?