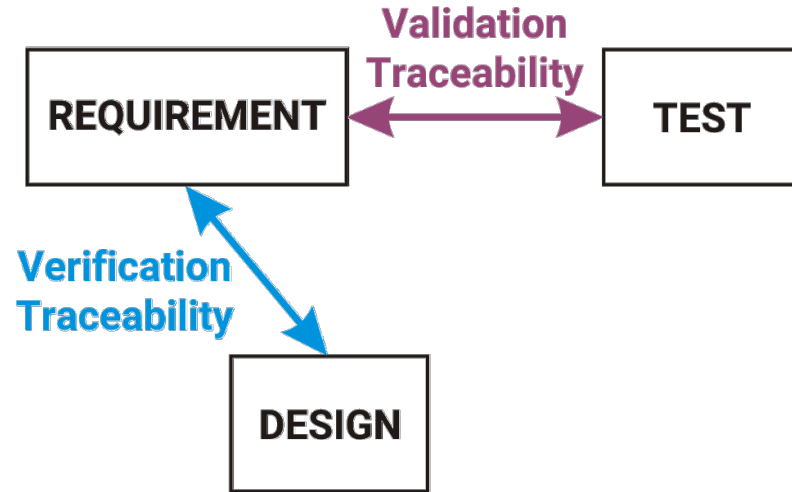# Traceability

There is never enough time to do it right the first time … but there is always enough time to do it over.

Traceability

■ **Anti-Patterns:**
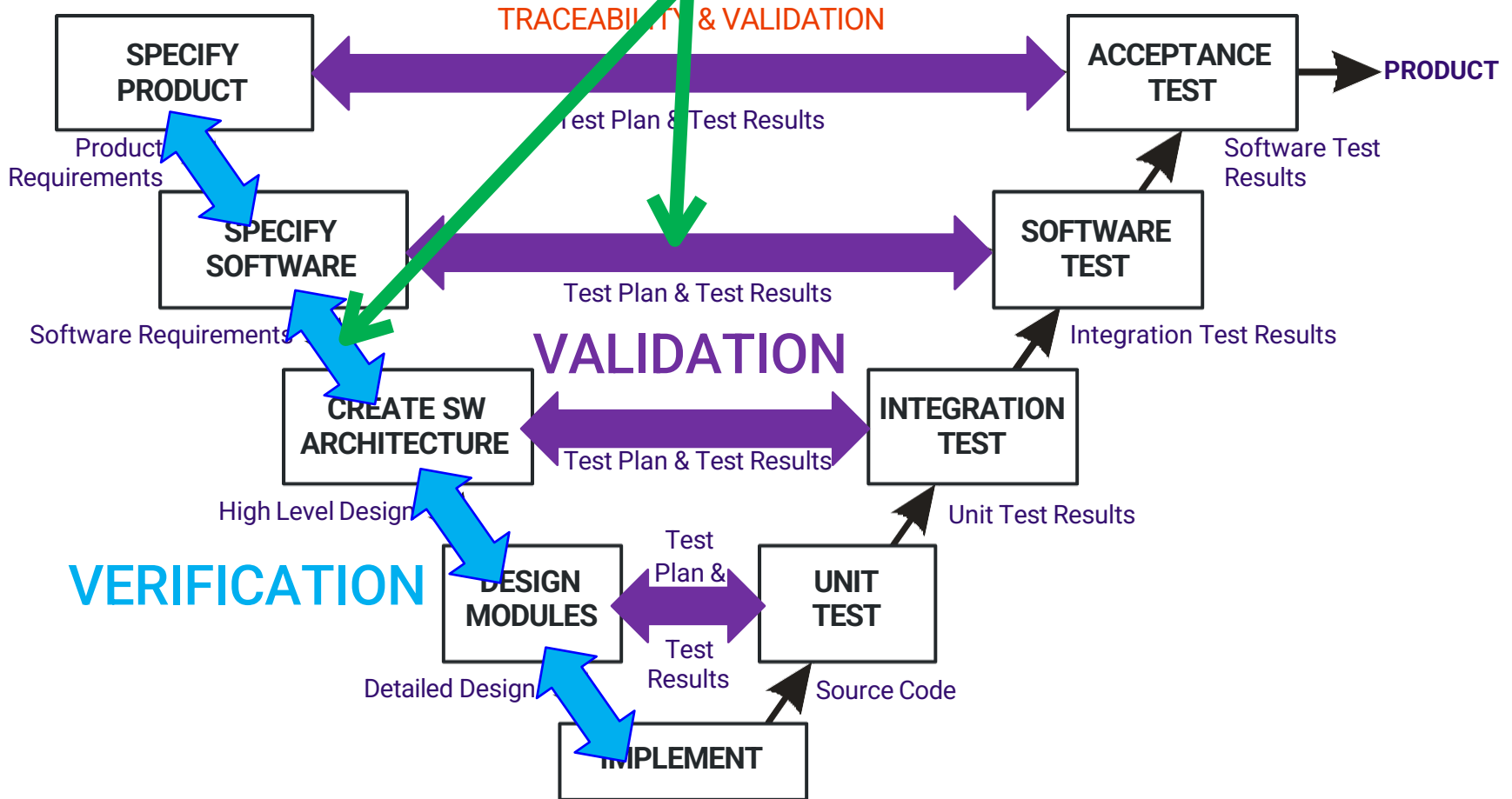- **Tests don't map to requirements**
- **Requirements aren't tested**
- **Reqts/design elements missing**
- **Gold plating (extra functionality)**

■ **Traceability**
- Creating something traces to a quality check on the result
  - <u>Verification:</u> you did something the way you said you'd do it
  - <u>Validation:</u> the thing you created behaves the way it should
- Ensure nothing left out; nothing added that shouldn't be there

YOU ARE HERE

TRACEABILITY & VALIDATION

SPECIFY PRODUCT → ACCEPTANCE TEST → PRODUCT

Product Requirements

Test Plan & Test Results

Software Test Results

SPECIFY SOFTWARE → SOFTWARE TEST

Software Requirements

Test Plan & Test Results

Integration Test Results

VALIDATION

CREATE SW ARCHITECTURE → INTEGRATION TEST

High Level Design

Test Plan & Test Results

Unit Test Results

VERIFICATION

DESIGN MODULES → UNIT TEST

Detailed Design

Test Plan & Test Results

Source Code

IMPLEMENT

3

Traceability Examples

- **Design traceability**
  - Requirement ➜ design ➜ implementation
  - Requirement ➜ test
- **SQA traceability**
  - Confirm process is being carried out
  - Process step ➜ document/artifact ➜ quality metric
- **Safety analysis traceability**
  - Confirm all hazards successfully mitigated
  - Hazard ➜ requirement ➜ mitigation ➜ validation
- **Defect traceability**
  - Ensure that all bugs are fixed
  - Bug report ➜ defect identified ➜ fix task ➜ code check-in ➜ regression test
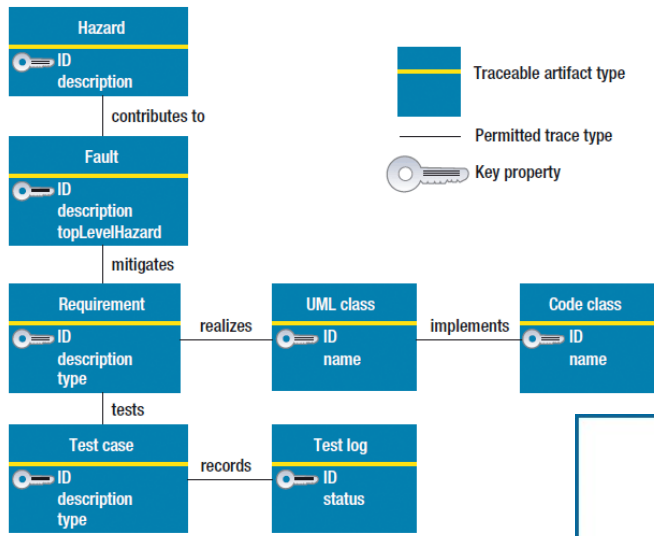
Requirements-to-Statecharts Traceability

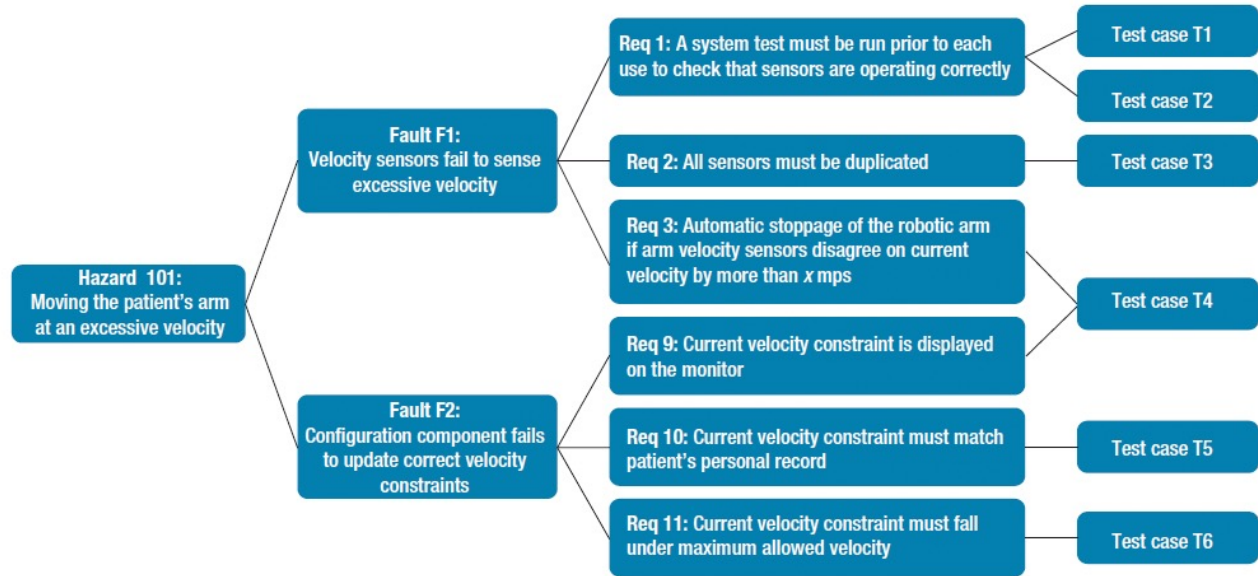|	Requirements					
States	R2.1	R2.2	R2.3	R2.4a	R2.4b	R2.5
S2.1 IDLE	x		x			x
S2.2 EMPTY	x	x				x
S2.3 VEND	x			x	x	
S2.4 FLASH_OFF	x			x	x	
S2.5 FLASH_ON	x			x	x	
Transitions						
T2.1				x	x	x
T2.2			x			
T2.3		x				
T2.4				x		
T2.5				x		
T2.6		x				
T2.7			x			

REQUIREMENTS TRACEABILITY MATRIX

Project Name: Online Flight Booking Application

Business Requirements Document BRD		Functional Requirements Document FSD			Test Case Document
Business Requirement ID#	Business Requirement / Business Use case	Functional Requirement ID#	Functional Requirement / Use Case	Priority	Test Case ID#
BR_1	Reservation Module	FR_1	One Way Ticket booking	High	TC#001 TC#002
		FR_2	Round Way Ticket		TC#003 TC#004
		FR_3	Multicity Ticket booking	High	TC#005 TC#006
BR_2	Payment Module	FR_4	By Credit Card	High	TC#007 TC#008
		FR_5	By Debit Card	High	TC#009
		FR_6	By Reward Points	Medium	TC#010 TC#011

Traceability for Hazard Mitigation

Mader et al. 2013:
http://doi.ieeecomputersociety.org/10.1109/MS.2013.60

Hazard
- ID
- description

Fault
- ID
- description
- topLevelHazard

Requirement
- ID
- description
- type

UML class
- ID
- name

Code class
- ID
- name

Test case
- ID
- description
- type

Test log
- ID
- status

Traceable artifact type
Permitted trace type
Key property

contributes to
mitigates
realizes
implements
tests
records

Hazard 101:
Moving the patient's arm at an excessive velocity

Fault F1:
Velocity sensors fail to sense excessive velocity

Fault F2:
Configuration component fails to update correct velocity constraints

Req 1: A system test must be run prior to each use to check that sensors are operating correctly

Req 2: All sensors must be duplicated

Req 3: Automatic stoppage of the robotic arm if arm velocity sensors disagree on current velocity by more than x mps

Req 9: Current velocity constraint is displayed on the monitor

Req 10: Current velocity constraint must match patient's personal record

Req 11: Current velocity constraint must fall under maximum allowed velocity

Test case T1
Test case T2
Test case T3
Test case T4
Test case T5
Test case T6

Traceability Best Practices

- **Trace everything in design package**
  - Even simple traceability checks can find problems
    - Gold plating: design item not traced to a requirement
  - Everything in design has an ID tag for traceability
    - Map left and right sides of V to each other
    - Map each layer of V upward and downward
  - Trace changes to see what else they affect
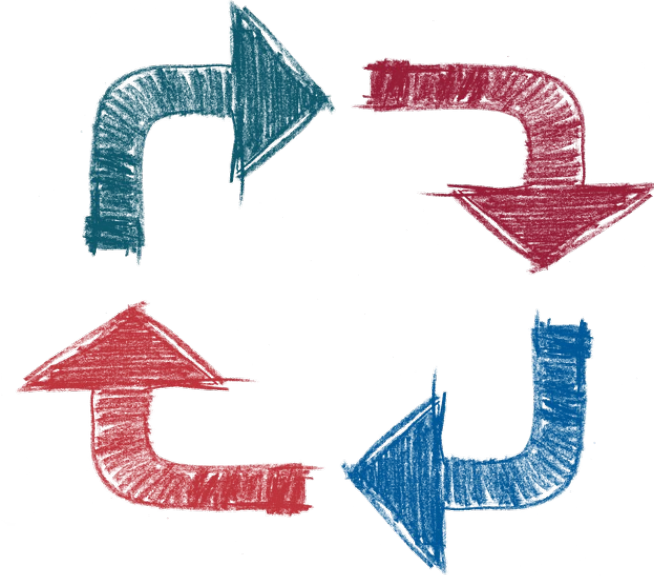
- **Traceability pitfalls**
  - Making granularity of trace IDs too big causes problems
  - Re-numbering breaks auto-generated document sections used as trace IDs
  - Don't use the wrong tool
    - Spreadsheets don't scale to big projects
    - Big project tools might be overkill where a spreadsheet approach will do

Disclaimer

This lecture contains materials from:

- Philip Koopman - CMU