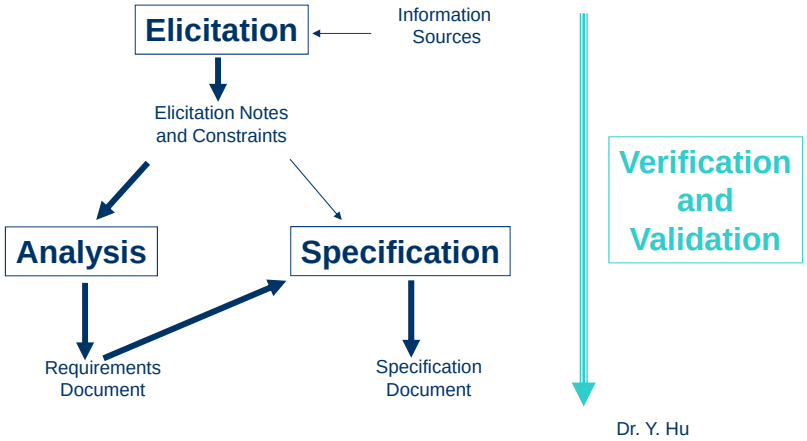
SENG 471

Software Requirements Engineering

Verification and Validation (V&V)

Refresh: RE Process

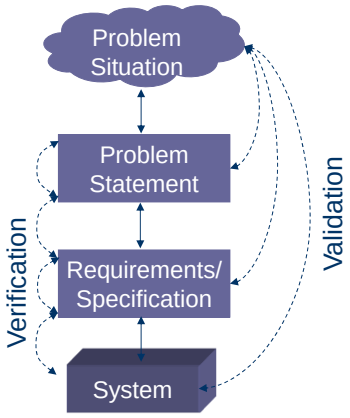


Dr. Y. Hu

What are the objectives of V&V?

Verifications & Validation (V&V)

- Verification → the system **right**?
  - requirements models ←→ one another?
  - delivered system ←→ what is said to do?
- Validation → the **right** system?
  - the problem statement ←→ the real problem?
  - we ←→ the needs of all the stakeholders?

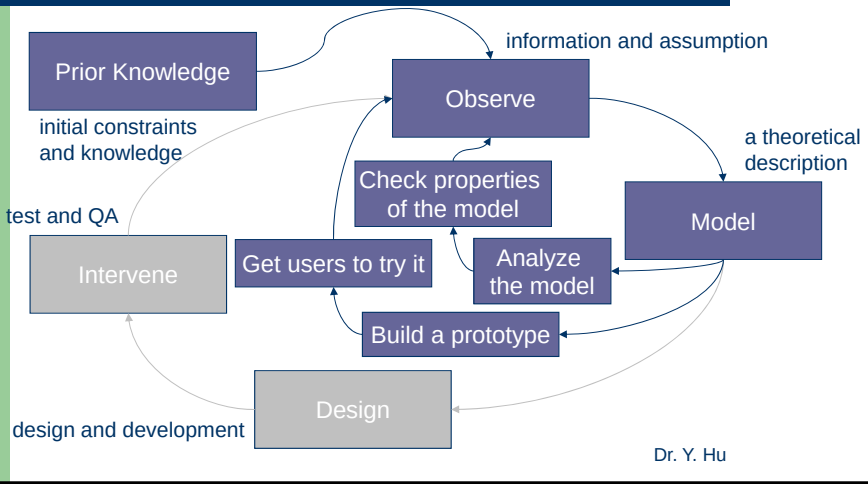


3

Dr. Y. Hu

- Requirements models are theories about the world.
- Design is tests of those theories.

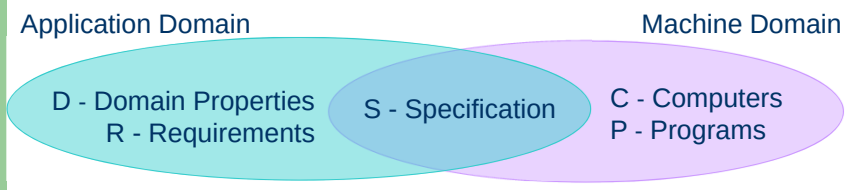
Cycle for RE



6

Dr. Y. Hu

Refresh: V&V Criteria



Two verification criteria:

- P, C ➔ S ?
- S, D ➔ R ?

Two validation criteria:

- all necessary R ?
- all relevant D ?

V&V Example

- Requirement **R**:
  - "During landing, forward thrust shall only be disabled when the aircraft is moving on the runway."
- Domain Properties **D**:
  - Wheel pulses are on if and only if wheels are turning.
  - Wheels are turning if and only if the aircraft is moving on the runway.
- Specification **S**:
  - Forward thrust shall be enabled if and only if wheel pulses are on.
- Verification ➔ S, D entails R ?
- Validation ➔ Did we miss any?

V&V Activities

- Reviews → Walkthroughs, inspections, etc.
- Software testing → Not applicable to RE.
- Formal methods → Use mathematics to prove that the requirements are consistent.
- Consistency checking → Verify consistency between models
- Prototyping → Present a prototype to the stakeholders to confirm its expected behaviors.
- Requirements tracing → Trace each requirement back to its source.

V&V Activities - Reviews

- (Fagan) Inspections - formal
  - used to improve quality of the development process
  - collect defect data to analyze the quality of the process
  - written output
  - train junior staff and transferring expertise
- Walkthroughs - informal
  - developer technique used by development teams to improve quality of product
  - focus on finding defects
- Management reviews
  - Used to provide confidence that the requirements are sound
  - Attended by management and sponsors (customers)
  - Often just a "dog-and-pony show"

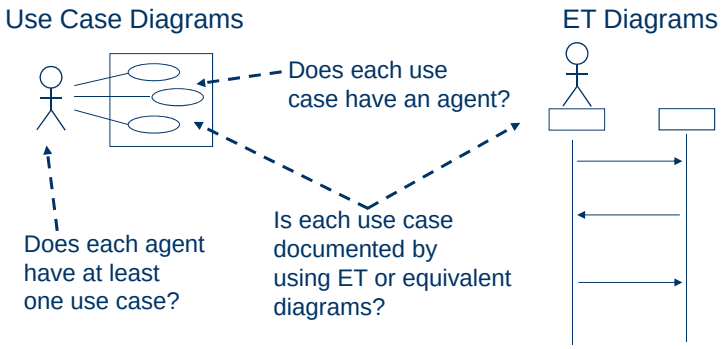
➢ Review the SRS with stakeholders to validate.

V&V Activities - Consistency

- BPMN diagrams (Activity diagrams)
  - All activities of a business?
- SADT diagrams (DFD, Use Case diagrams)
  - A flow of data is associated with activities, and vice versa?
  - Each case has a user and is documented?
- ER diagrams (Class diagrams)
  - A diagram captures all entities in other diagrams?
  - Every entity has its attributes?
- ET diagrams (Sequence diagrams)
  - Each agent is in a ER diagram and has messages?
- SCR tables (SM, Statechart, R-net diagrams)
  - Each diagram capture (the states of) an entity?
  - Each state is identified by attribute values?
  - Each transition have a trigger event?

11

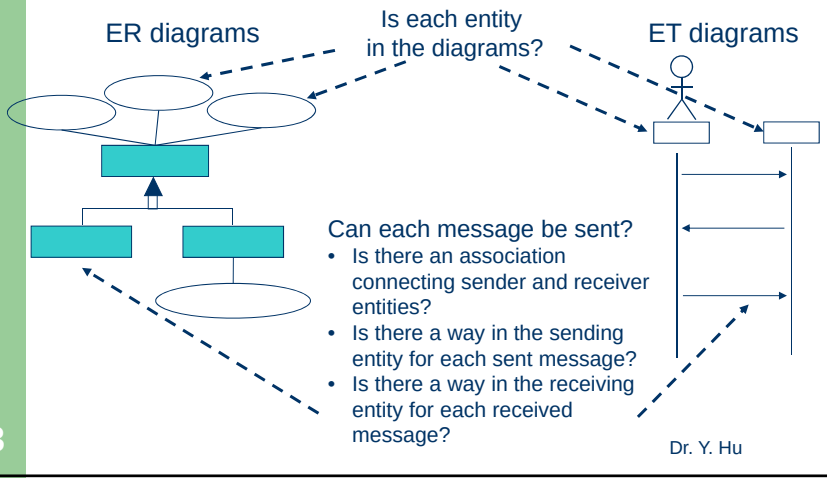
Consistency Checking – Example 1



12

Dr. Y. Hu

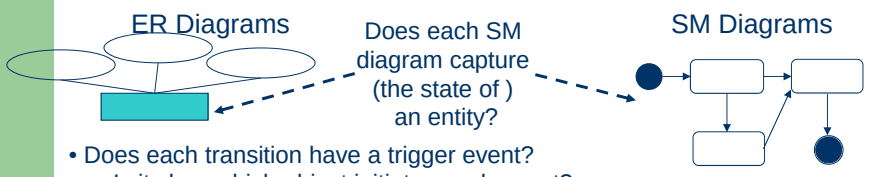
Consistency Checking – Example 2



Can each message be sent?

- Is there an association connecting sender and receiver entities?
- Is there a way in the sending entity for each sent message?
- Is there a way in the receiving entity for each received message?

Consistency Checking – Example 3



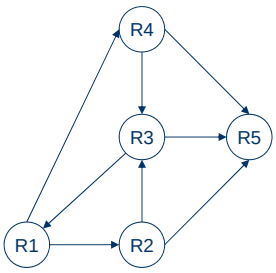
- Does each transition have a trigger event?
  - Is it clear which object initiates each event?
  - Is each event an operation for that entity in the ER diagram?
- Does each state represent a distinct combination of attributes?
  - Is it clear which combination of attributes?
  - Are all those attributes shown on the ER diagram?
- Are there ways in the ER diagram for each transition?
  - …a way that will update attributes for the new state?
  - …ways that will test any conditions on the transition?
  - …ways that will carry out any actions on the transition?

V&V Activities - Prototyping

- "A software prototype is a partial implementation constructed primarily to enable customers, users, or developers to learn more about a problem or its solution."
- "Prototyping is the process of building a working model of the system."
- Approaches
  - Presentation prototypes
  - Exploratory prototypes
  - Breadboards or experimental prototypes
  - Evolutionary prototypes ("operational" or "pilot")

V&V Activities - Tracing

Traceable items	R1	R2	R3	R4	R5
R1	0	1	0	1	0
R2	0	0	1	0	1
R3	1	0	0	0	1
R4	0	0	1	0	1
R5	0	0	0	0	0



Tracing - forward

- **Forward traceability**:
  stakeholders ➔ requirements specification
- Traceability matrix:

ID	Requirements	Forward Traceability
S2	Users shall process retirement claims	R10, R11, R12
S3	Users shall process survivor claims	R13

18

Dr. Y. Hu

Tracing - backward

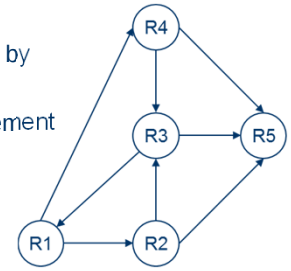
- **Backward traceability**:
  requirements specification ➔ stakeholders.
- Traceability matrix:

ID	Requirements	Backward Traceability
R10	The system shall accept requirement data.	
R11	The system shall calculate the amount of retirement.	
R12	The system shall calculate point-to-point travel time.	
R13	The system shall calculate the amount of survivor annuity.	

19

Requirements Traceability

- From ISO/IEEE-STD:
  - Forward traceability
    - Trace forward to all documents spawned by the SRS.
    - Facilitation of referencing of each requirement in future documentation.
    - Each requirement has a unique name or reference number.
  - Backward traceability
    - Trace backward to previous stages of the SRS.
    - The origin of each requirement should be clear.



Traceability - Importance

- Verification and Validation
  - Assess adequacy of test suite
  - Assess conformance to requirements
  - Assess completeness, consistency, impact analysis
  - Detect requirements conflicts
  - Check consistency of decision making across the lifecycle
- Maintenance
  - Assess change requests
  - Trace design rationale
- Process visibility
  - See how the software was developed
  - Provide an audit trail
- Management
  - Change management
  - Risk management
  - Control of the development process

Traceability - Current Practice

- Coverage:
  - link between requirements at different levels
  - link from requirements forward to designs, code, test cases
  - link back from designs, code, test cases to requirements
- Traceability process
  - Assign each requirement/specification a unique id#
  - Identify linkages
  - Use tables to record linkages in a document
  - Use a traceability tool (database) for project wide traceability
  - Some software tools

Traceability - Current Limitations

- Informational problems
  - Tools fail to track useful traceability information
  - Inadequate pre-requirements traceability
- Lack of agreement…
  - …over the quantity and type of information to trace
- Informal communication
  - People attach great importance to personal contact and informal communication
  - But then the traceability database only tells part of the story!

Independent V&V

V&V by **separate contractors**

- Independent technical opinions
- About 5% ~ 15% of development costs
- Five-fold return on investment:
  - Errors found earlier, cheaper to fix, cheaper to re-test
  - Clearer specifications
  - Developer more likely to use best practices

Three types of independence

- Technical Independence:
  - Avoid analyst bias
  - Use different tools and techniques
- Financial Independence:
  - Separate cost and fund
  - No diverting resources when the thing gets tough
- Managerial Independence:
  - Separate responsibility
  - Decide when and where to focus the V&V effort

Dr. Y. Hu

25

Case Study - Mars polar lander

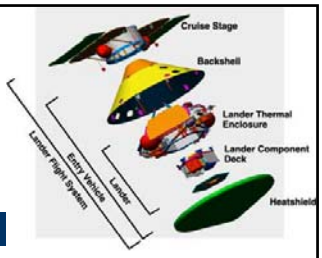
- Launched:
  - 3 Jan 1999
- Mission
  - Near South Pole
  - Water ice
- Fate:
  - No signal, after initial phase of descent
- Cause:
  - Most likely: premature engine shutdown

Dr. Y. Hu

26

* S/W = software

What Happened?



- Cause ➔ Transient signals on touchdown
  - S/W* accepts the signals for 2 timeframes.
- Factors ➔ System ignorance + Poor testing
  - No S/W requirements of the signals.
  - No unit-test on the signals.
  - No full-test after re-wiring sensors.
- Result ➔ Premature shutdown of engines
  - S/W signals touchdown.
  - Lander: ~40 m above surface; traveled at 13 m/s.
  - Impact velocity: ~ 22 m/s (no survive of the lander)
  - Nominal touchdown velocity: 2.4 m/s

27

Dr. Y. Hu

* Adapted from: "Report of the loss of the Mars Polar Lander and Deep Space 2 missions – JPL Special Review Board (Casani Report) - March 2000".

NASA Report*

SYSTEM REQUIREMENTS		FLIGHT SOFTWARE REQUIREMENTS
	3.7.2.2.4.2	Processing
1) The touchdown sensors shall be sampled at 100-Hz rate.	a.	The lander flight software shall cyclically check the state of each of the three touchdown sensors (one per leg) at 100 Hz during EDL.
The sampling process shall be initiated prior to lander entry to keep processor demand constant.	b.	The lander flight software shall be able to cyclically check the touchdown event state with or without touchdown event generation enabled.
However, the use of the touchdown sensor data shall not begin until 12 meters above the surface.	X	
2) Each of the 3 touchdown sensors shall be tested automatically and independently prior to use of the touchdown sensor data in the onboard logic.	c.	Upon enabling touchdown event generation, the lander flight software shall attempt to detect failed sensors by marking the sensor as bad when the sensor indicates "touchdown state" on two consecutive reads.
The test shall consist of two (2) sequential sensor readings showing the expected sensor status.	d.	The lander flight software shall generate the landing event based on two consecutive reads indicating touchdown from any one of the "good" touchdown sensors.
If a sensor appears failed, it shall not be considered in the descent engine termination decision.		
3) Touchdown determination shall be based on two sequential reads of a single sensor indicating touchdown.		

28

Recap

- V&V objectives
- V&V activities