

# **Evidence-based Development – coupling structured argumentation with requirements development**

*Jeremy.Dick@integrate.biz*



**based on paper ...**

***Paper:***

- “EVIDENCE-BASED DEVELOPMENT – COUPLING STRUCTURED ARGUMENTATION WITH REQUIREMENTS DEVELOPMENT”
- IET System Safety – Edinburgh, September 2012

***Experience:***

- application in large UK defence and civil nuclear projects





# agenda

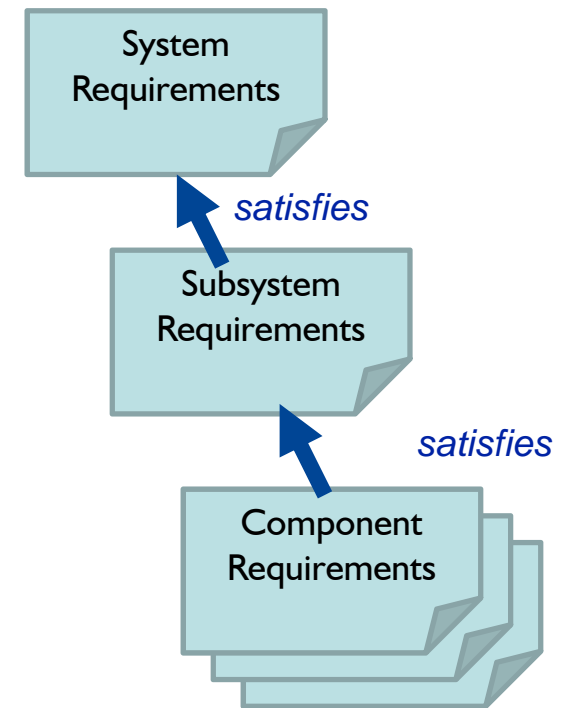
- requirements in brief – overview
- a shift in thinking – reviewing traceability
- traceability, rationale and evidence
- Evidence-based Development
- conclusion





# requirements in brief

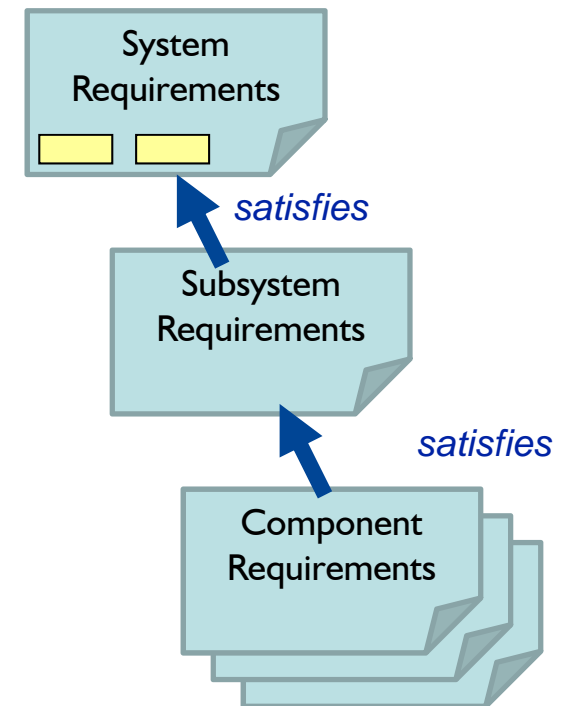
- manage requirements in layers





# requirements in brief

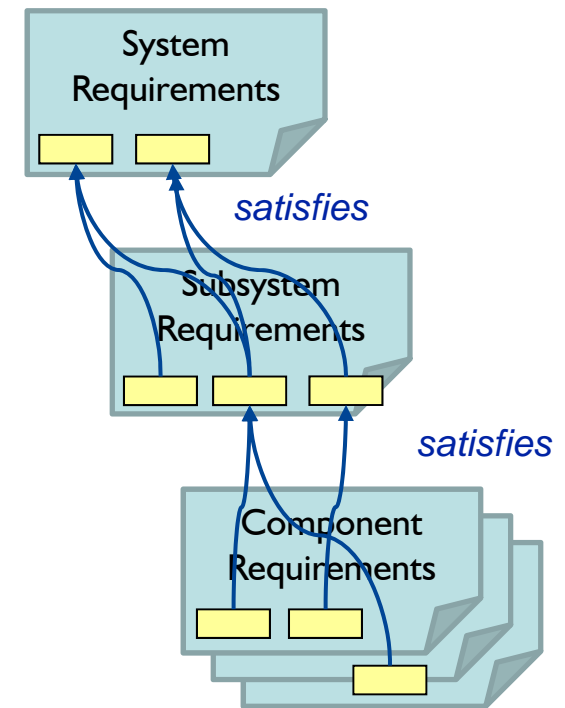
- manage requirements in layers
- express requirements as traceable statements in appropriate language





# requirements in brief

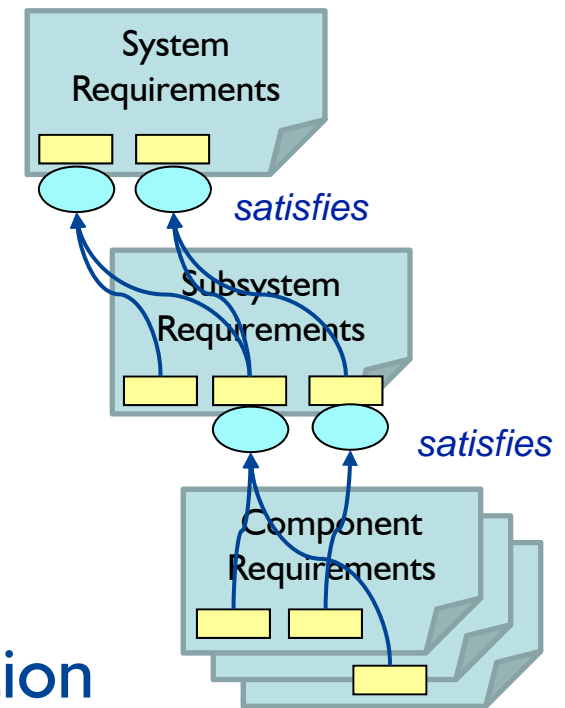
- manage requirements in layers
- express requirements as traceable statements in appropriate language
- systematically decompose requirements through the layers and record the relationships





# requirements in brief

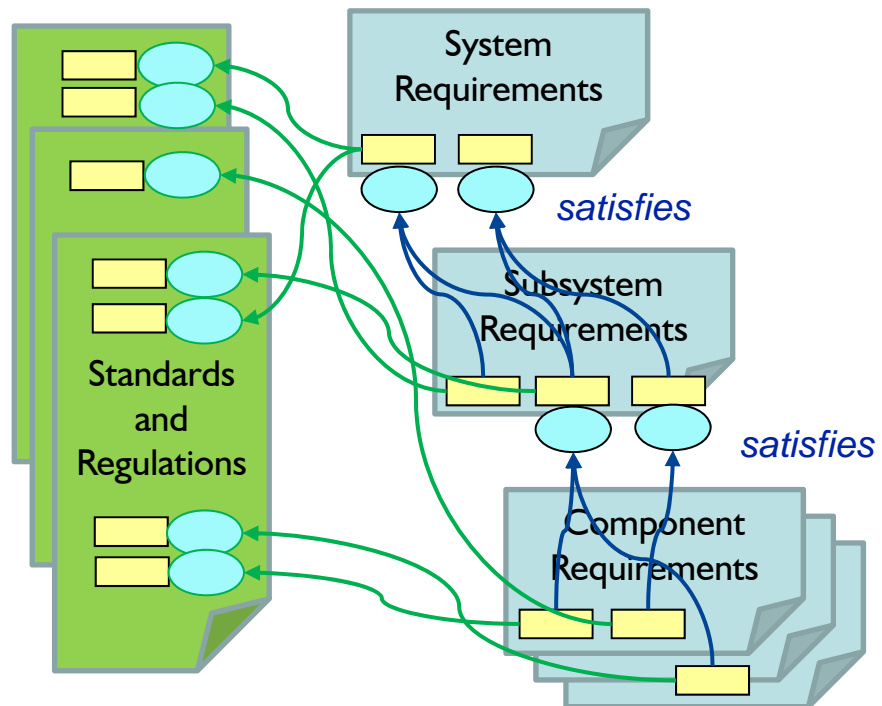
- manage requirements in layers
- express requirements as traceable statements in appropriate language
- systematically decompose requirements through the layers and record the relationships
- record the rationale for the decomposition – compliance information





# requirements in brief

- demonstrate compliance

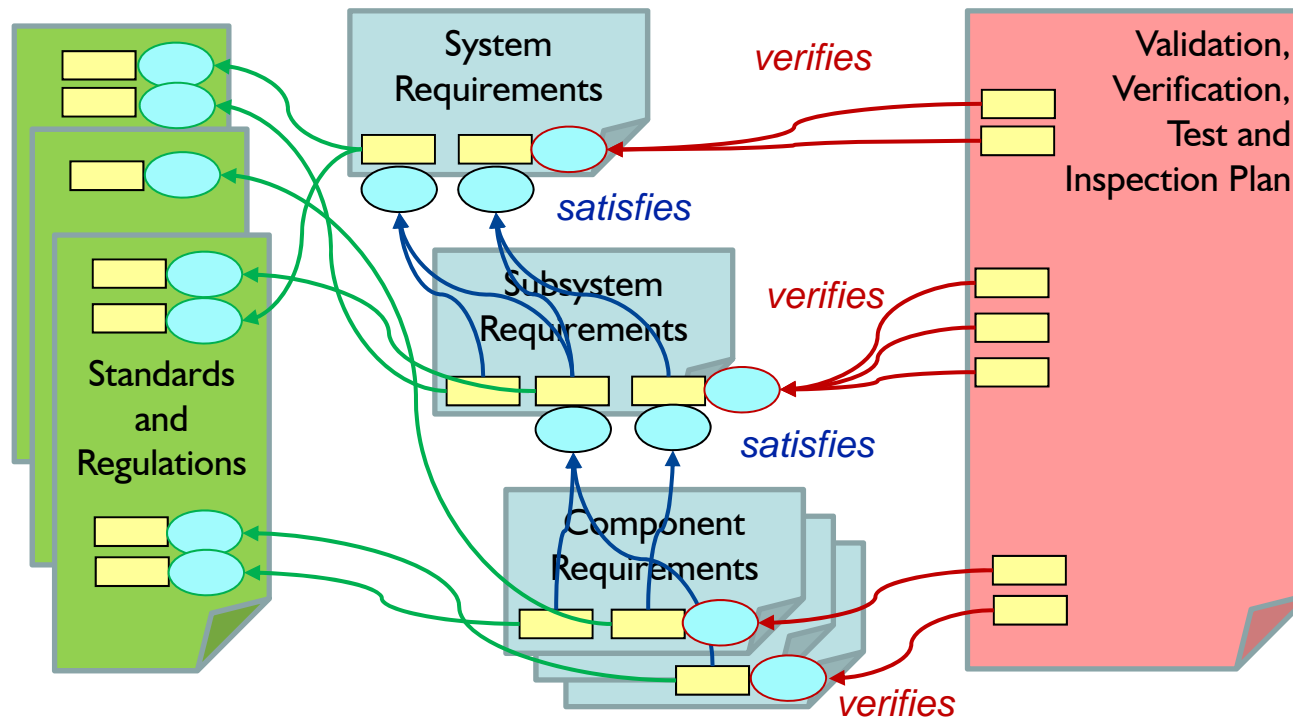






# requirements in brief

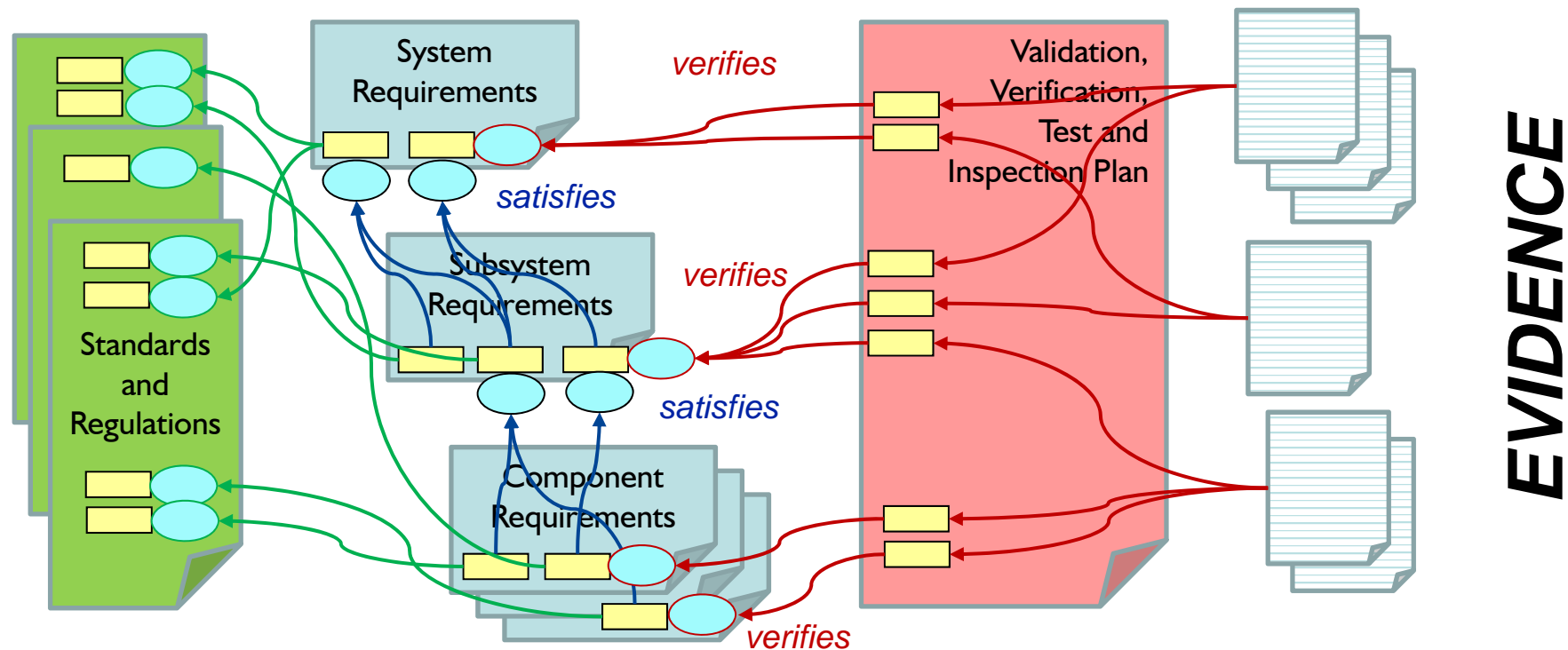
- plan tests against requirements





# requirements in brief

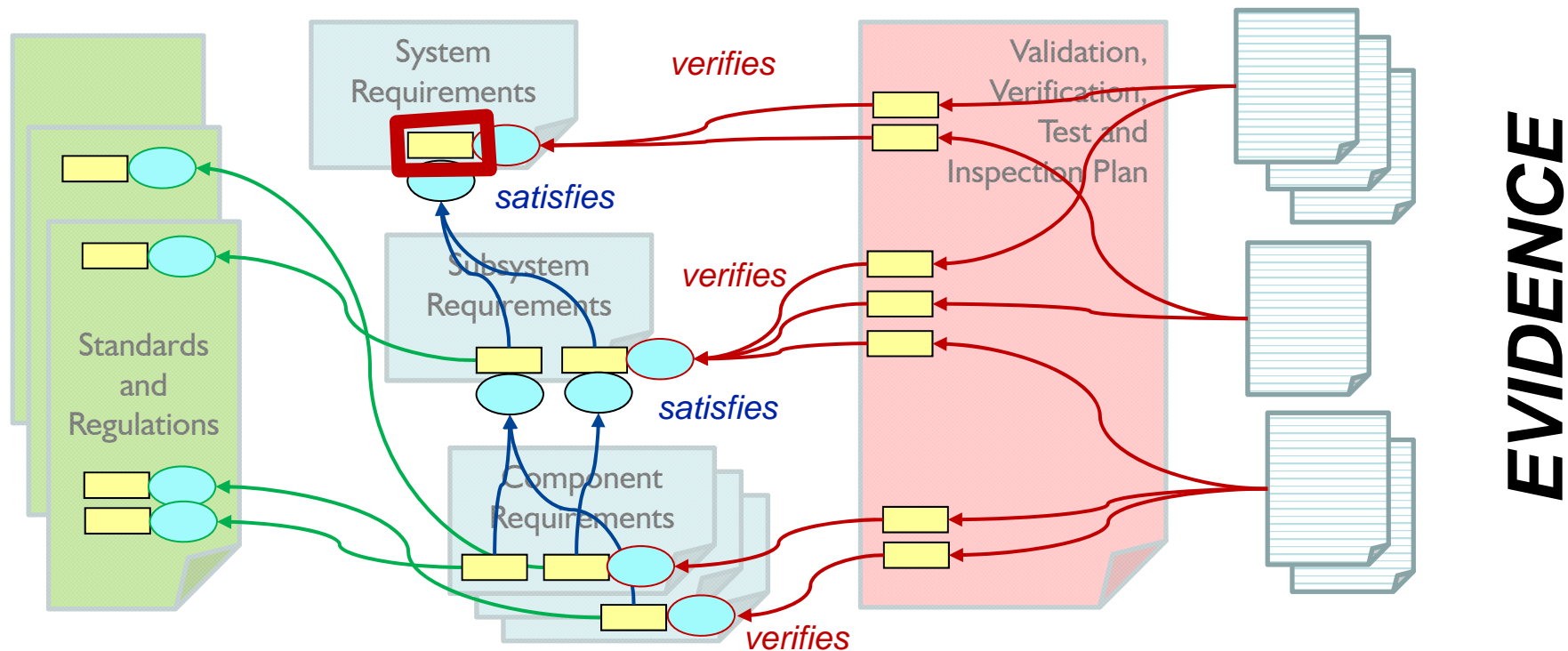
- collect results/evidence against test plans





# requirements in brief

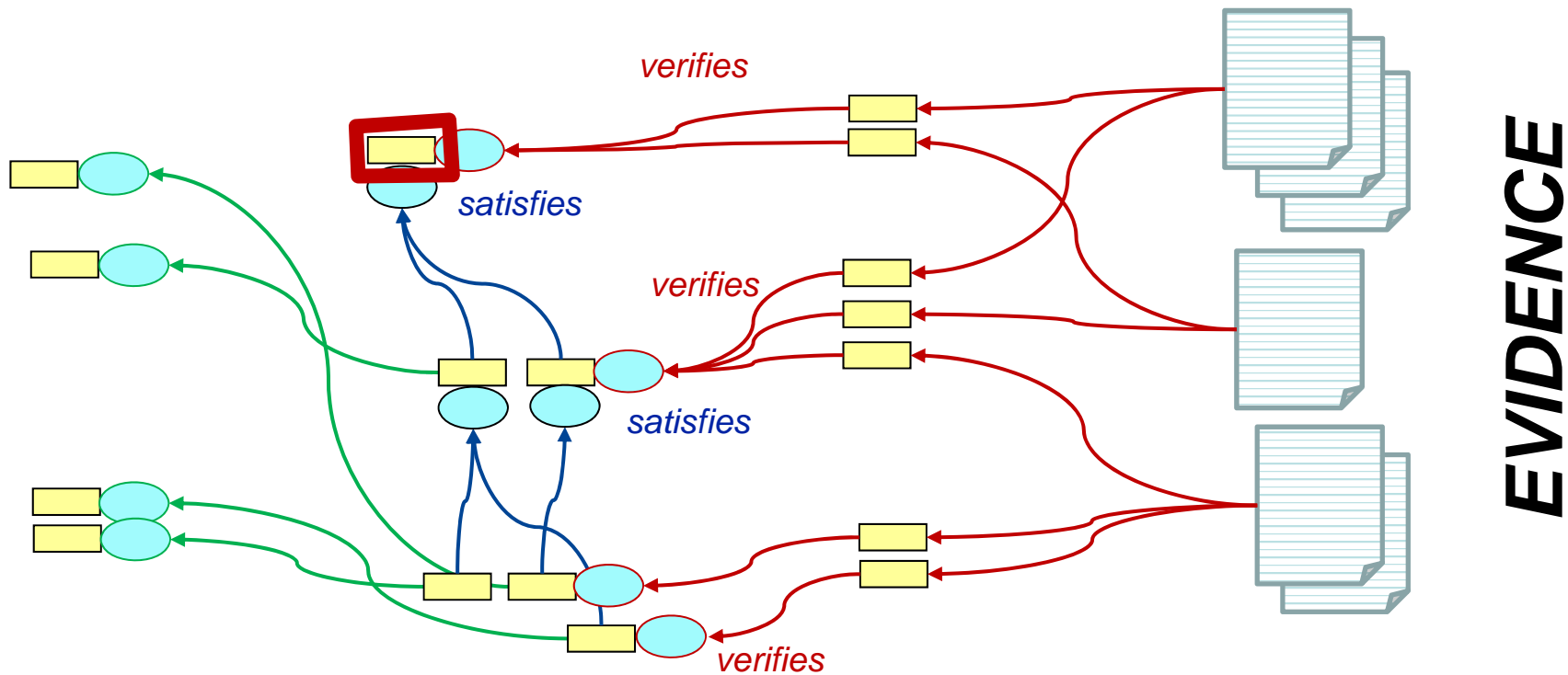
- manage change through impact analysis



**EVIDENCE**

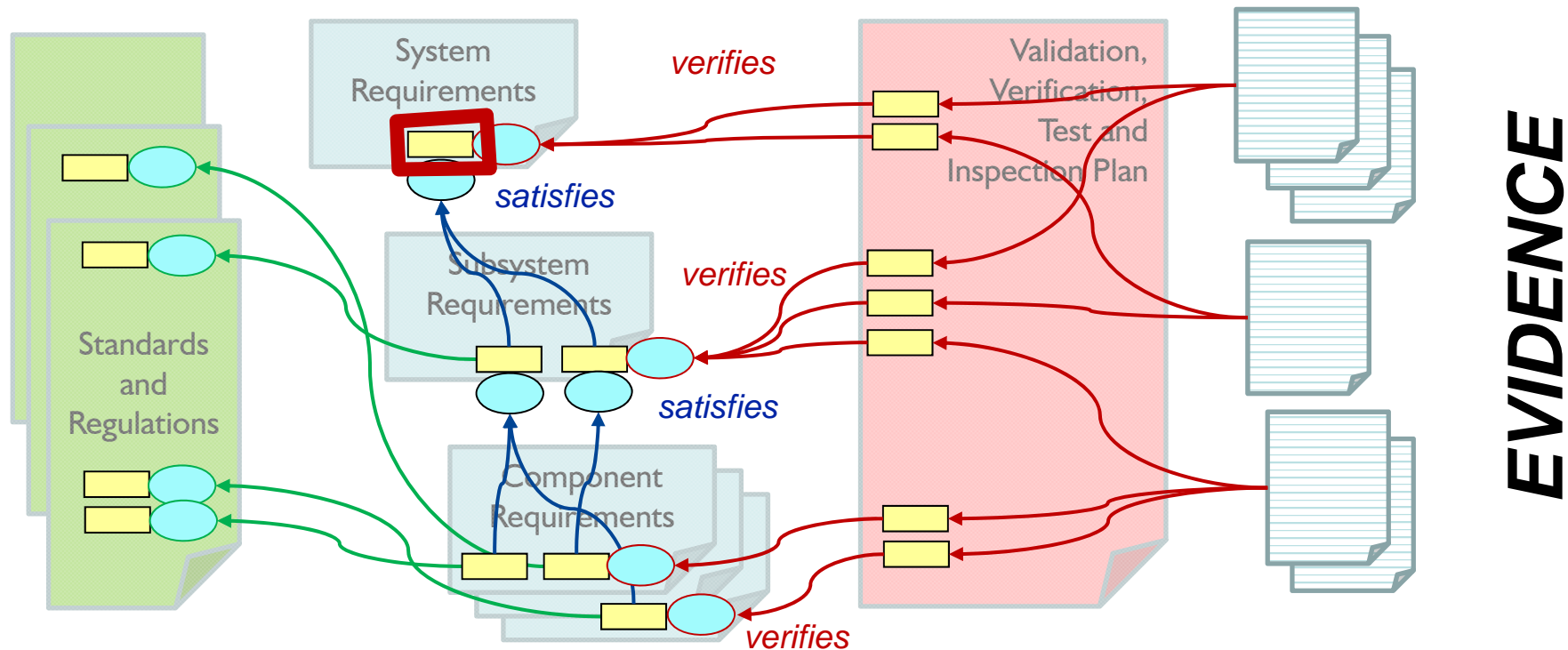


# assurance case for requirement





# assurance-based development or evidence-based development



**EVIDENCE**



# a shift in thinking

**Stakeholder  
Requirements**

**System  
Requirements**

**Subsystem  
Requirements**

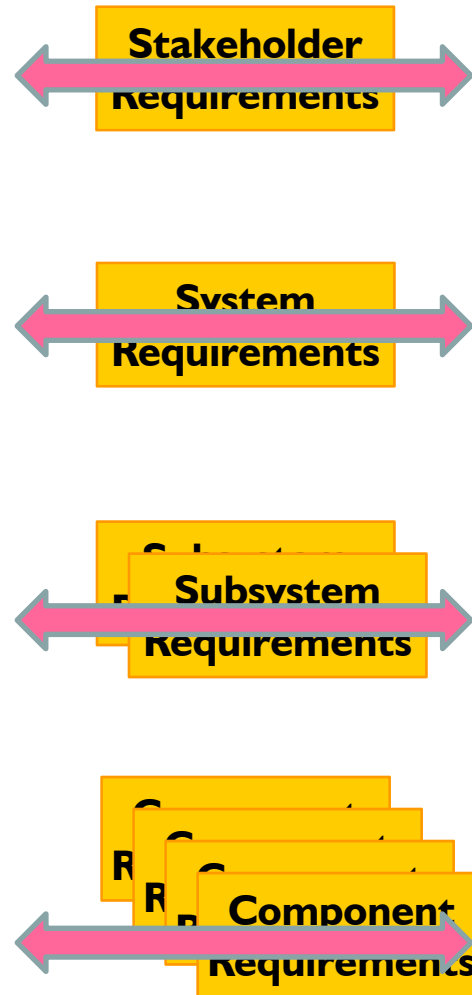
**Component  
Requirements**





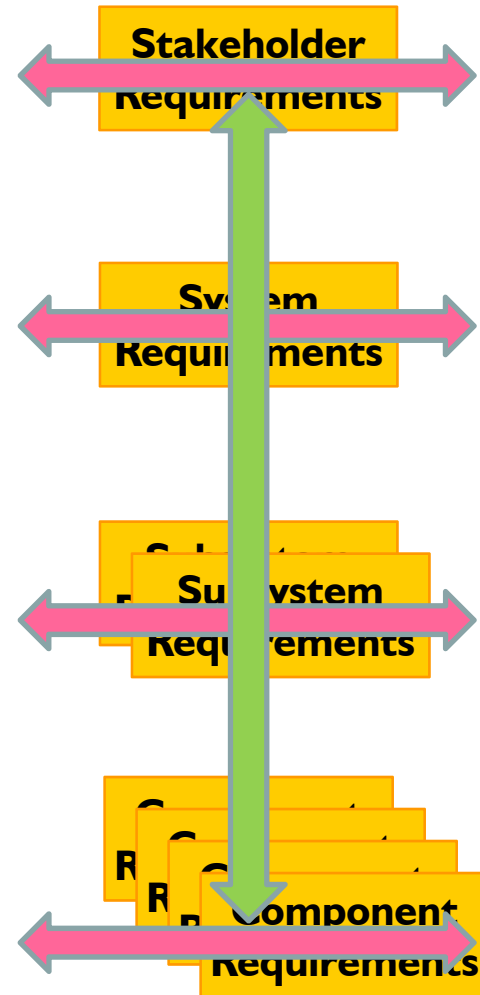
# a shift in thinking

- from reviewing requirements in isolation:
  - review one requirements document at a time





# a shift in thinking



- from reviewing requirements in isolation:
  - review one requirements document at a time
- to reviewing in context:
  - review relationship between requirements at multiple levels

***because most of the meaning is captured in the relationships, not just in the requirements***



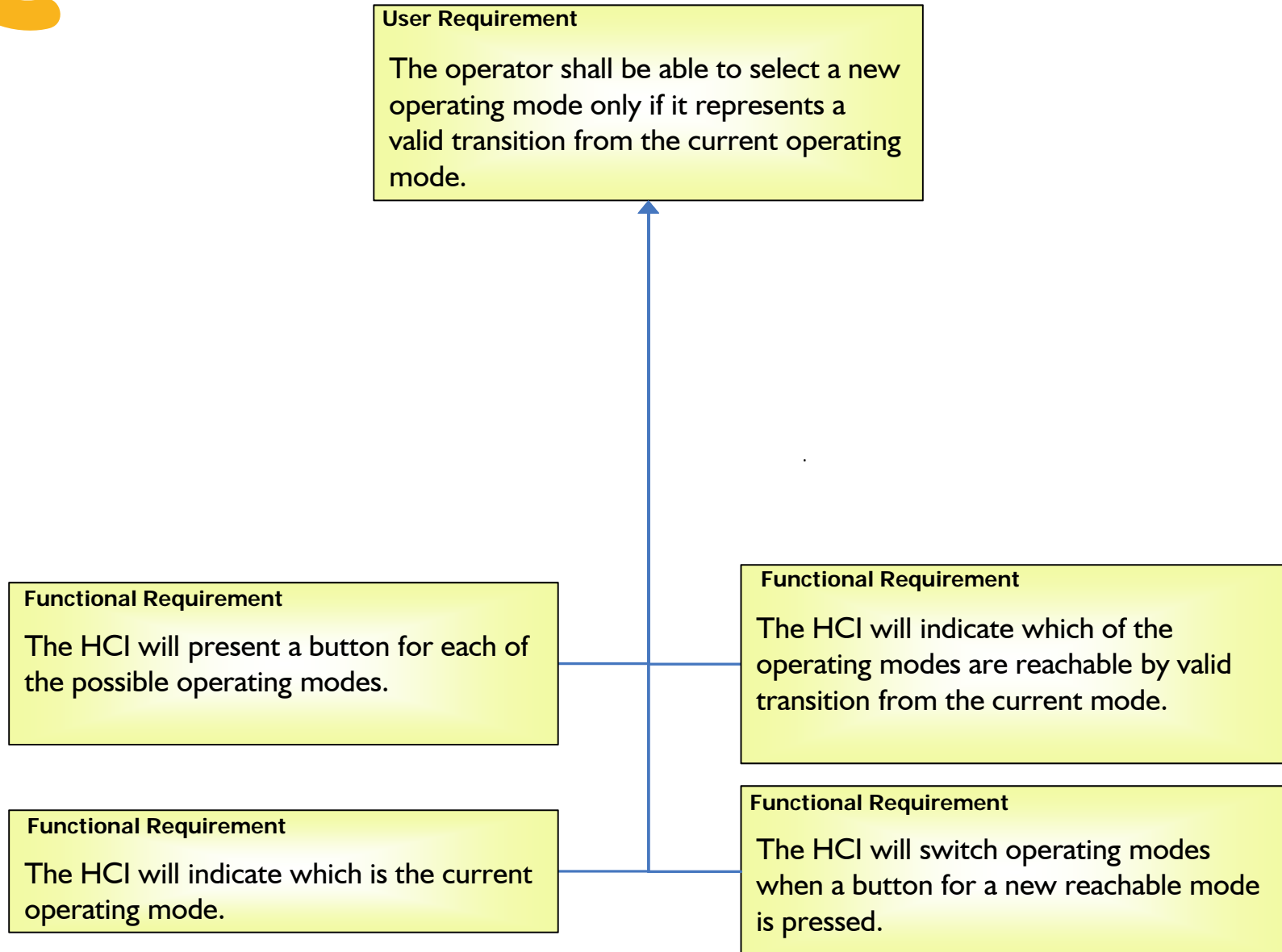


# traceability

- documenting how high-level requirements are transformed into low-level requirements
- understanding how requirements are satisfied, validated and verified
- explaining compliance against regulations
- ***connecting evidence to design***

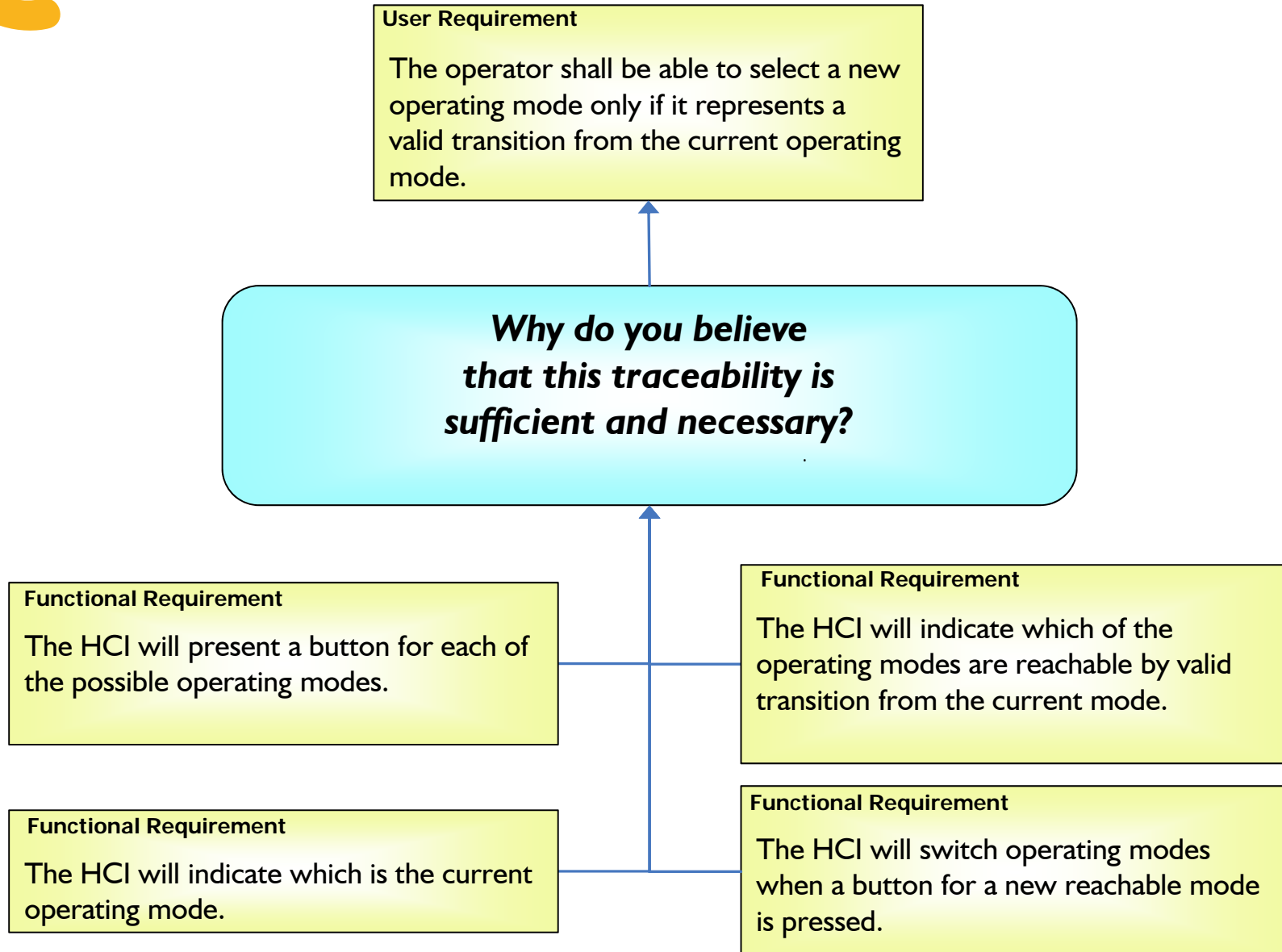


# tracing: satisfaction relationship



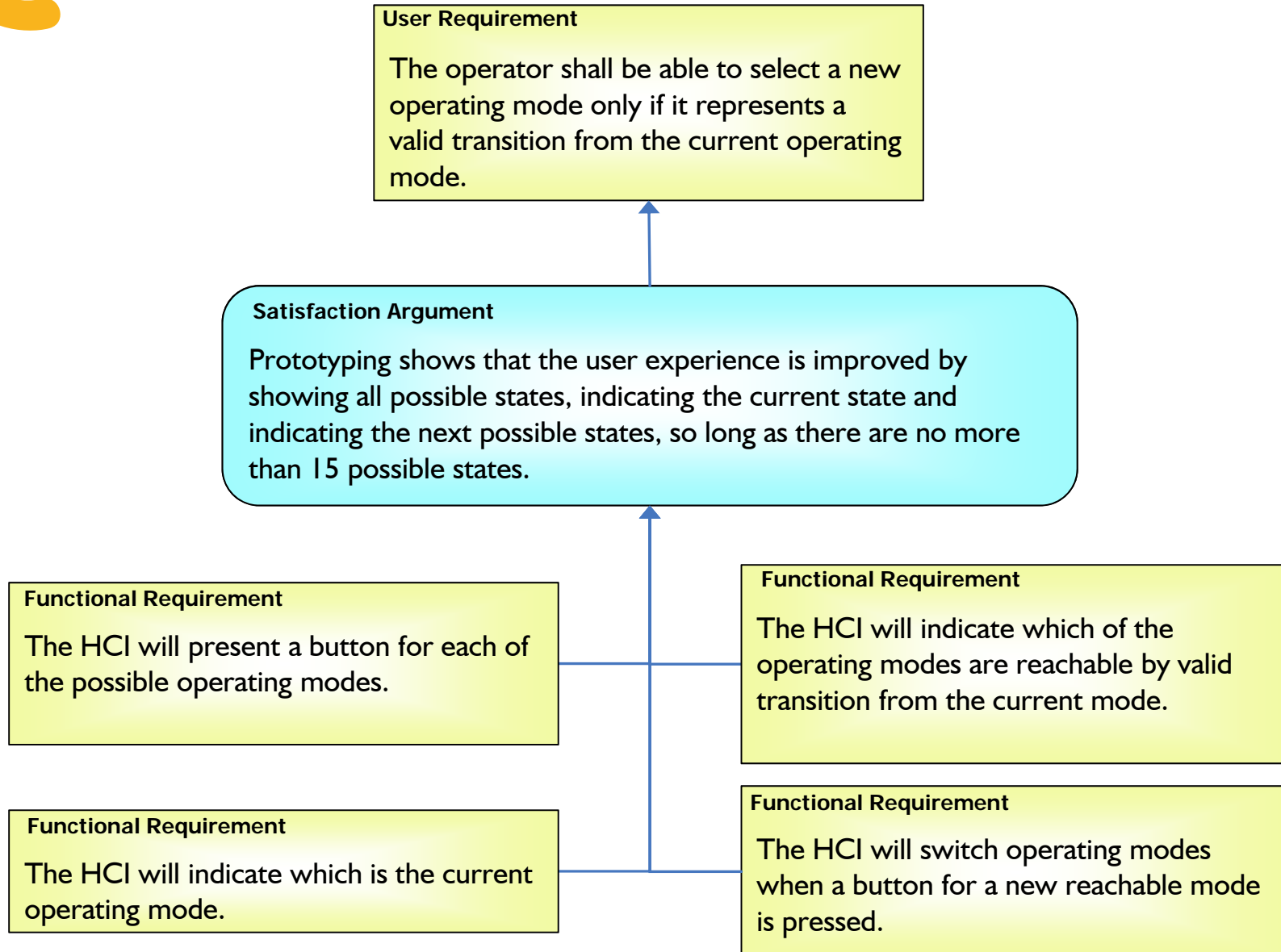


# satisfaction argument



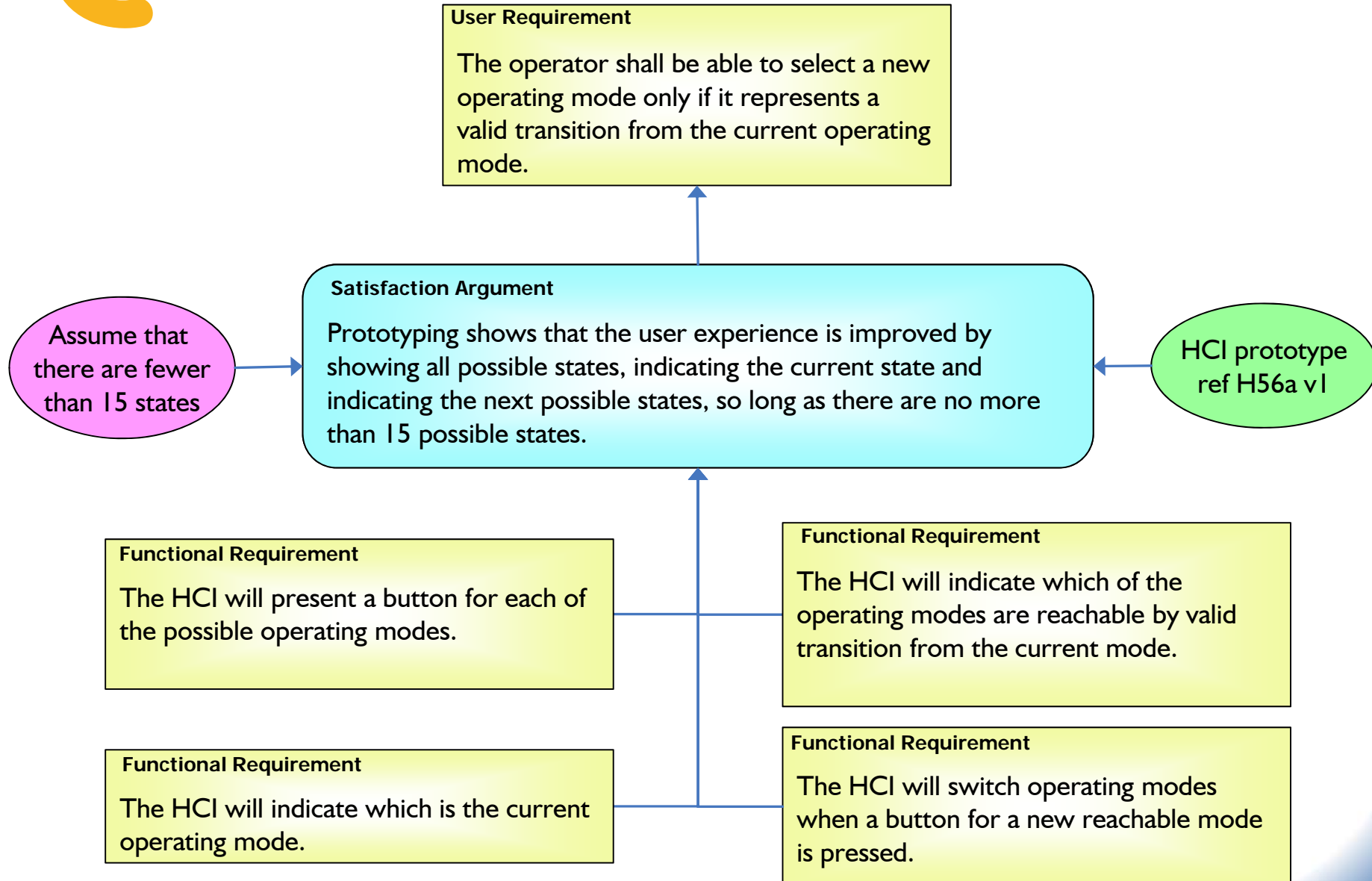


# satisfaction argument



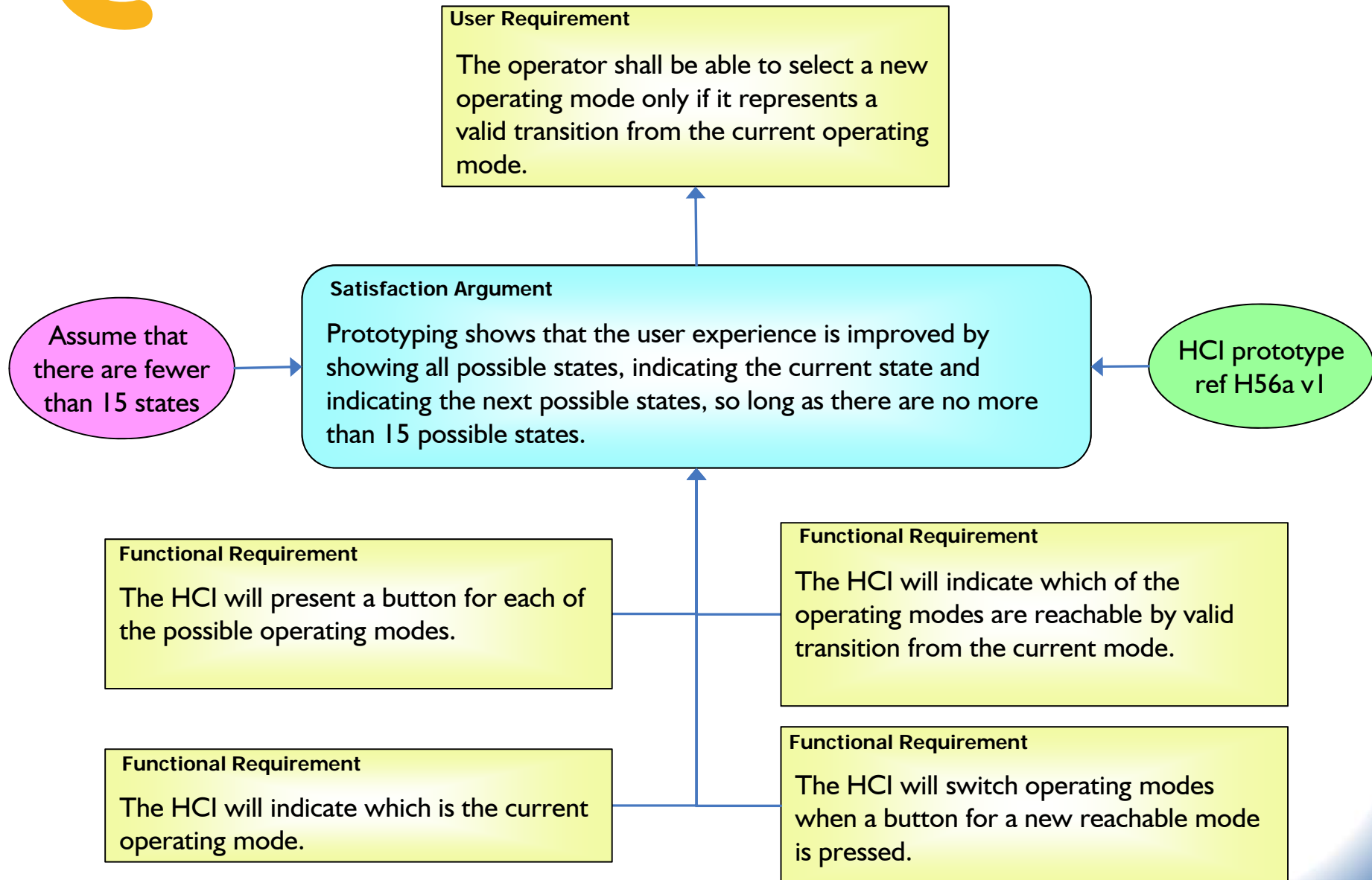


# structured argumentation



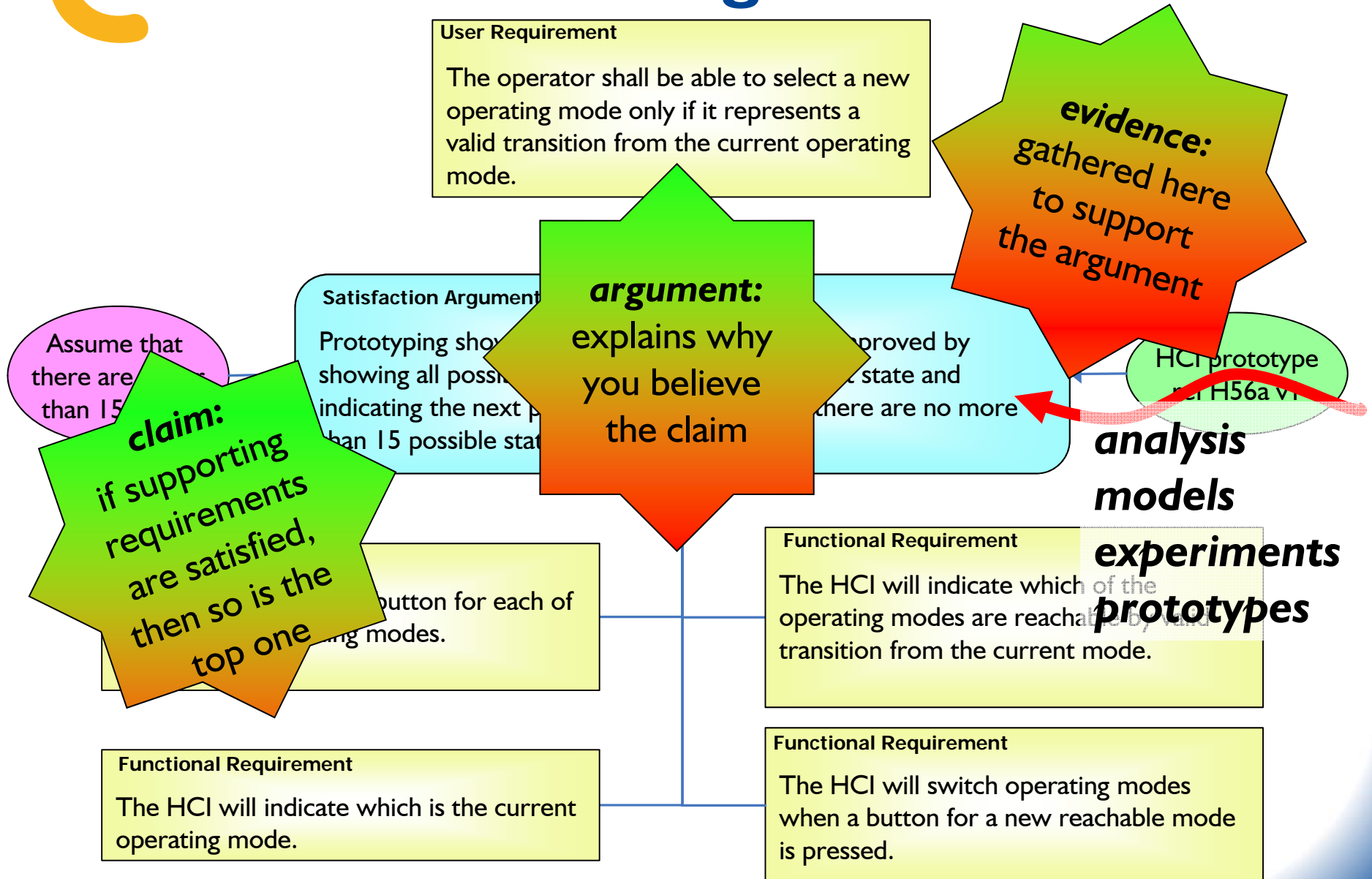


# Goal Structuring Notation?





# claim / argument / evidence





# tracing: verification relationship

**User Requirement**  
The operator shall be able to select a new operating mode only if it represents a valid transition from the current operating mode.

**Acceptance Test**  
Put the facility into a valid operating state, and attempt to select a new valid operating state.

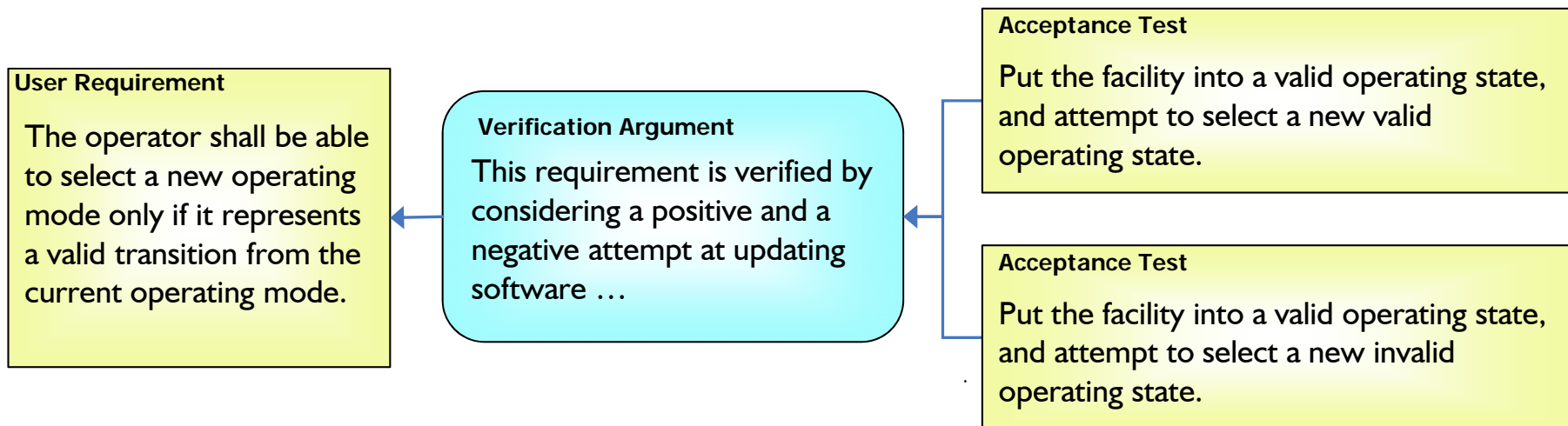
**Acceptance Test**  
Put the facility into a valid operating state, and attempt to select a new invalid operating state.





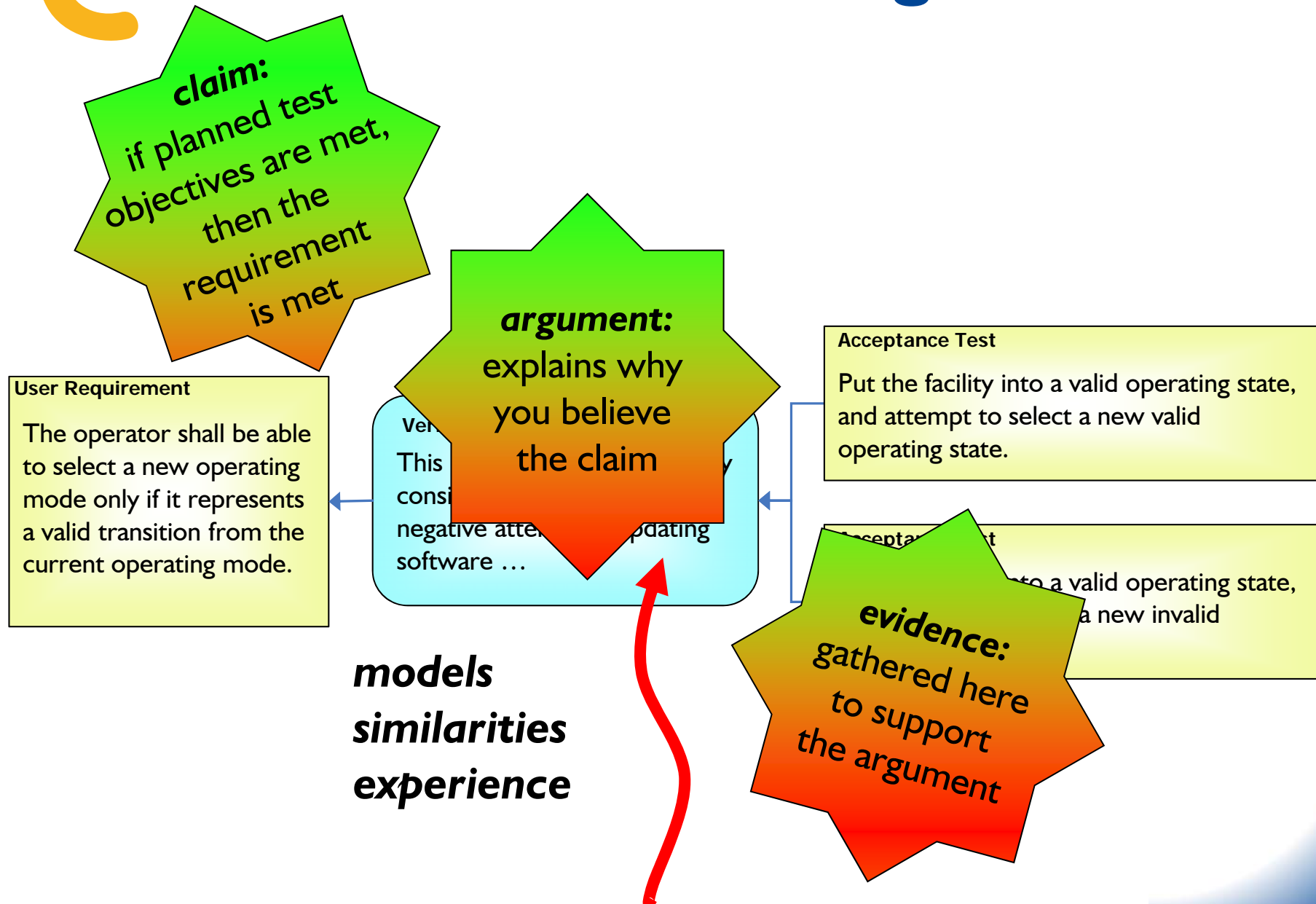


# verification argument



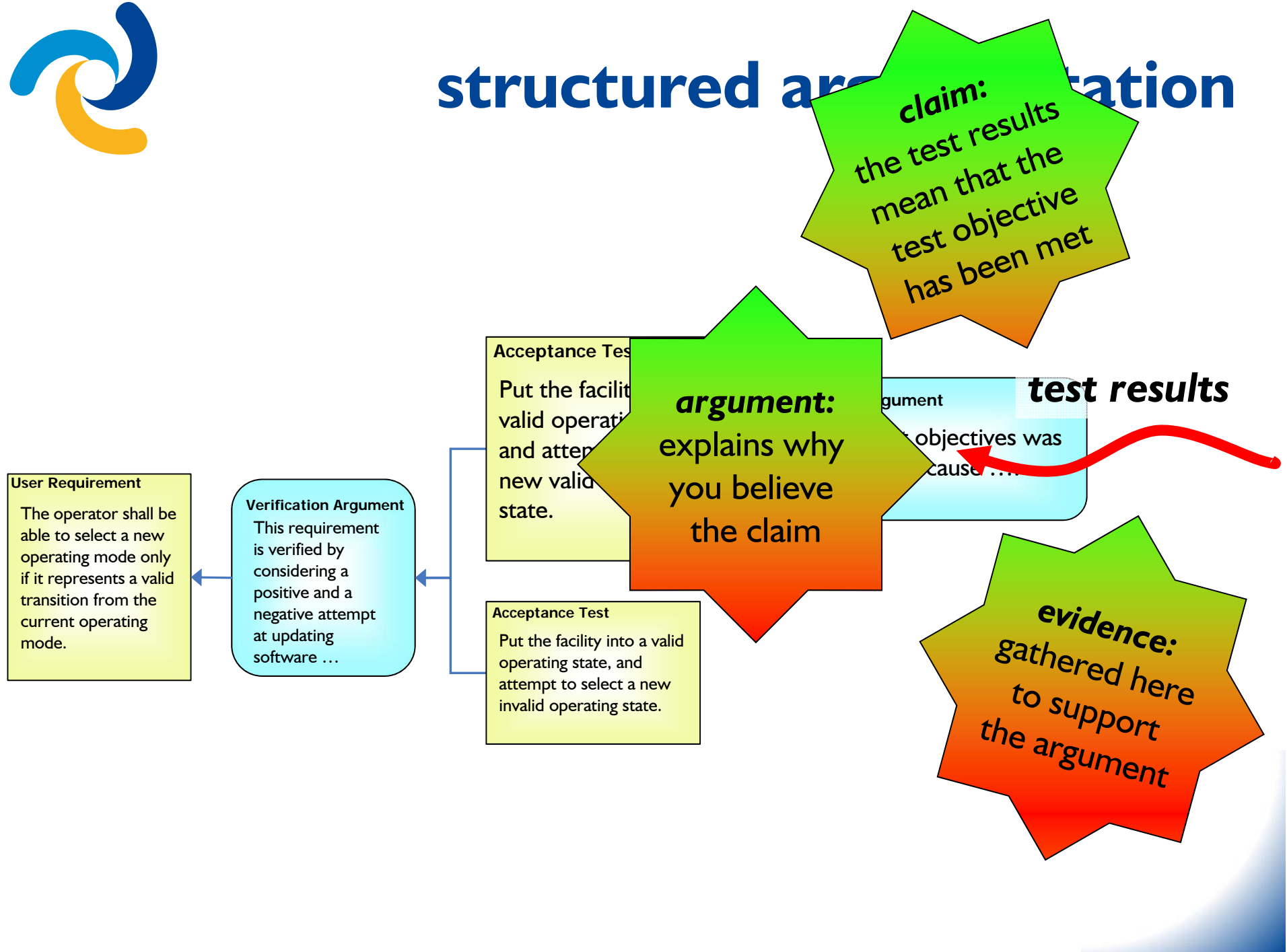


# structured argumentation





# structured argumentation





# Evidence-based Development

## *What is it?*

- framework for collecting evidence for the correctness of a system as you design the system
- uses requirements traceability as the structure for establishing arguments and supporting evidence
- extends the structured argument paradigm to cover all kinds of requirement – not just safety
- gives ownership of assurance to every engineer

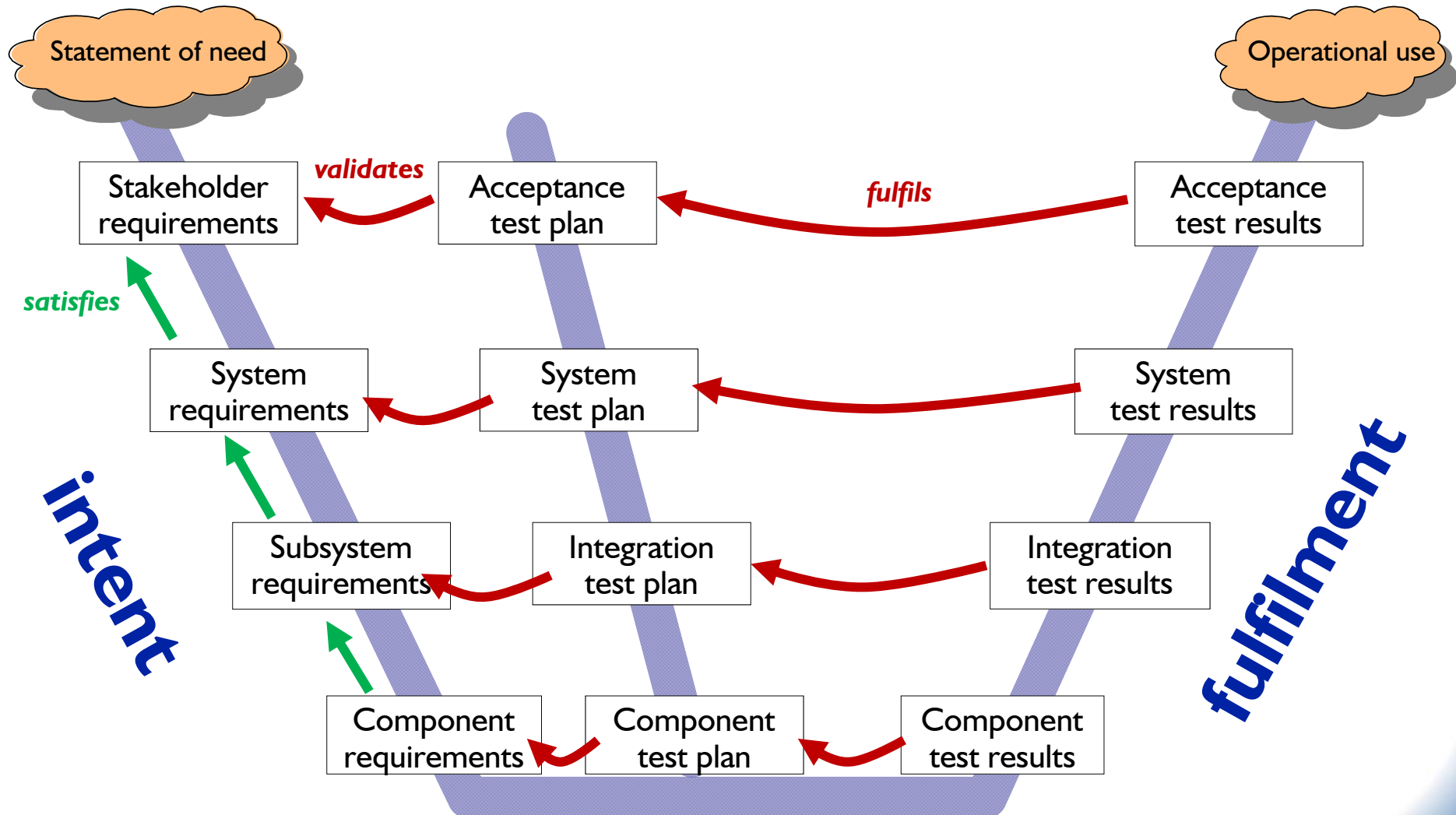


## objectives of EbD

- connect the assurance case connected to the design
  - the assurance case should not be an “after-thought”
- develop the assurance case early
  - in time to influence the design
  - in time to save costly rework late in the day
- apply a uniform approach to all aspects of assurance
  - address all kinds of requirements: function, performance, ease-of-use, reliability, safety, ...
  - have a single point of reference for structured argumentation

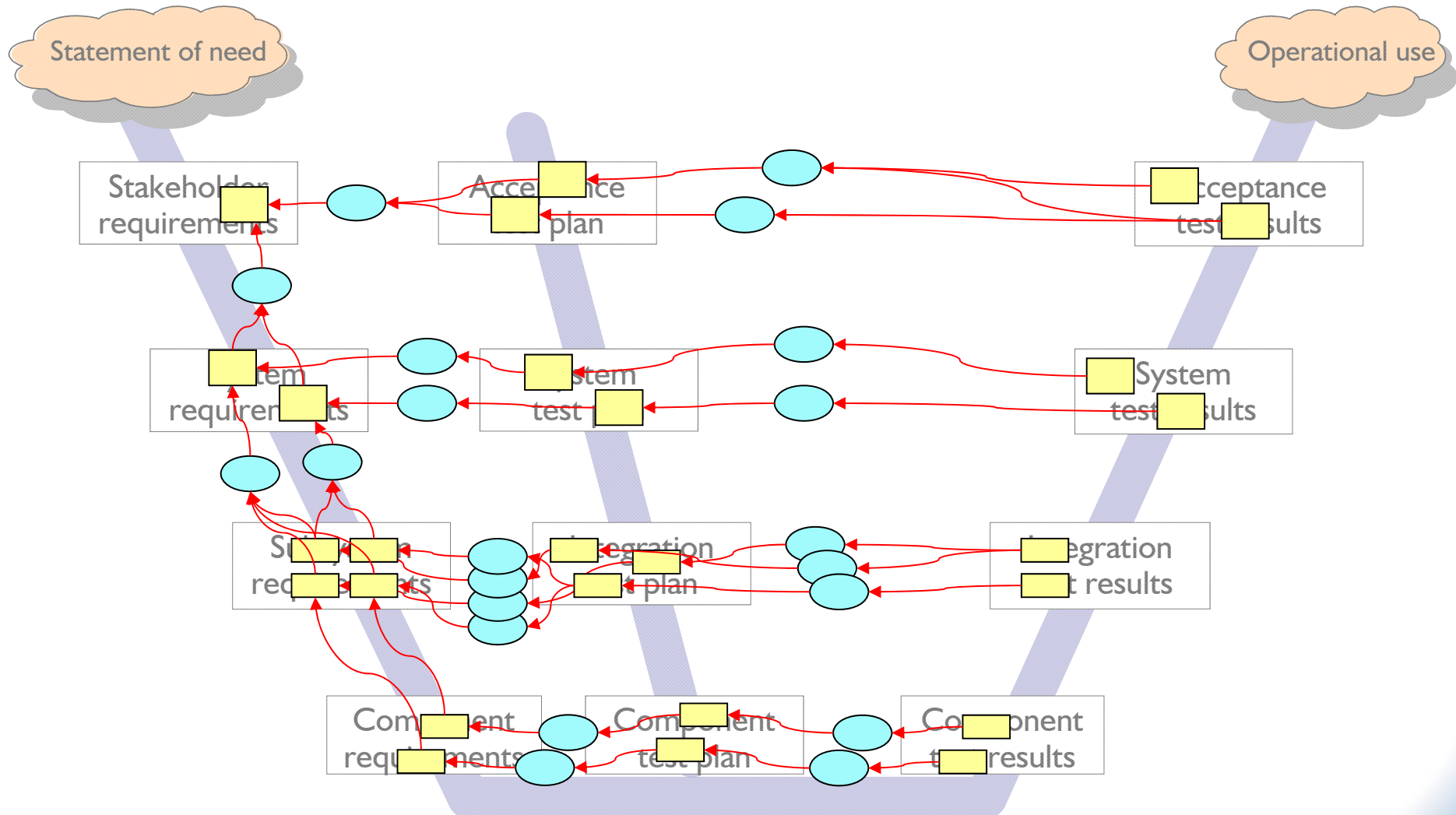


# relationships in the 'W' model



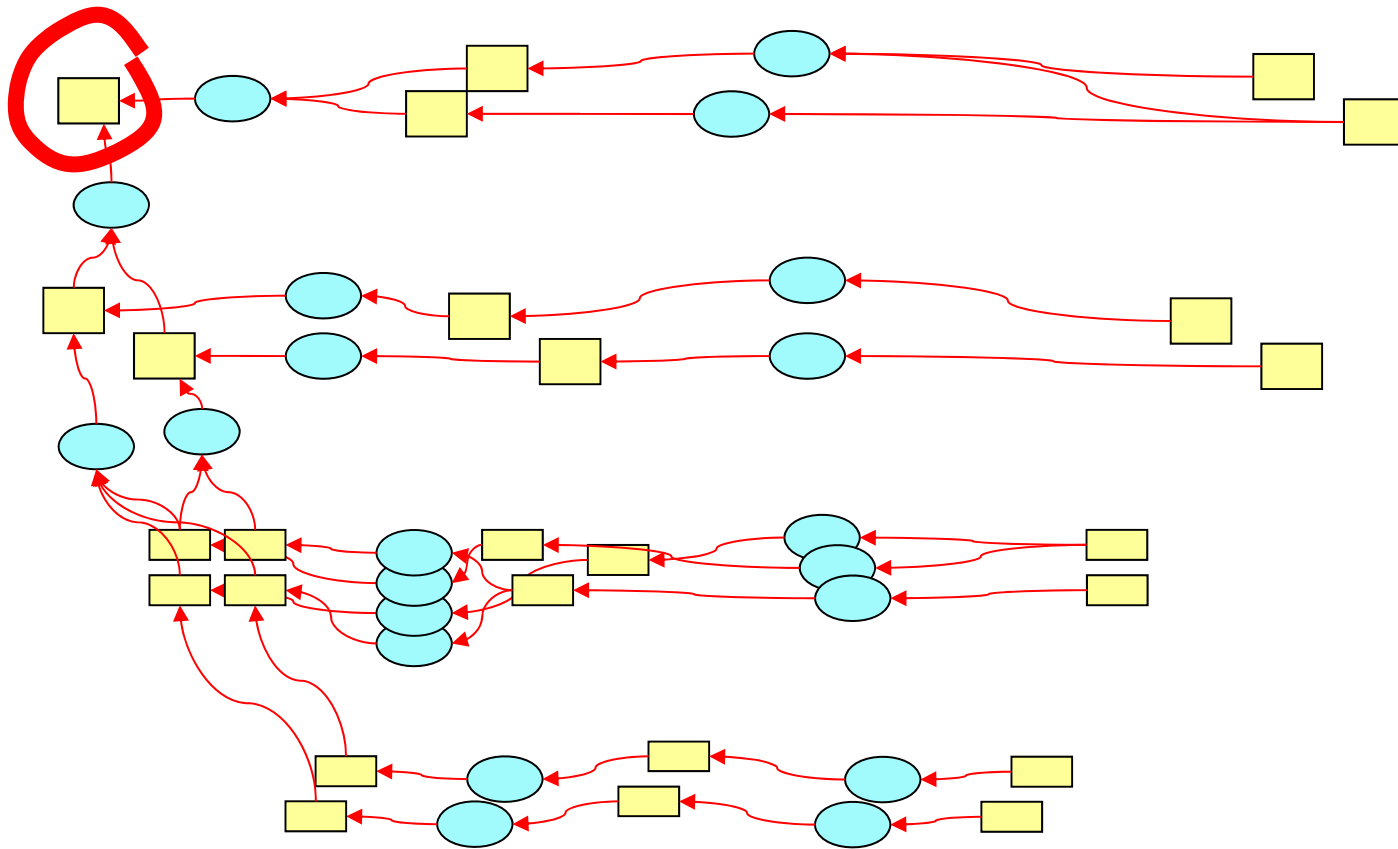


# progressive assurance





# single requirement assurance case







**testing mantra**

***test early, test often***

**because**

**the sooner you find defects,**

**the cheaper it is to fix**





# testing (V&V) covers

## *early:*

- design analysis
- design modelling
- design reviews

## *middle:*

- component tests
- factory tests
- integration tests

## *late:*

- systems tests
- operational tests
- acceptance tests

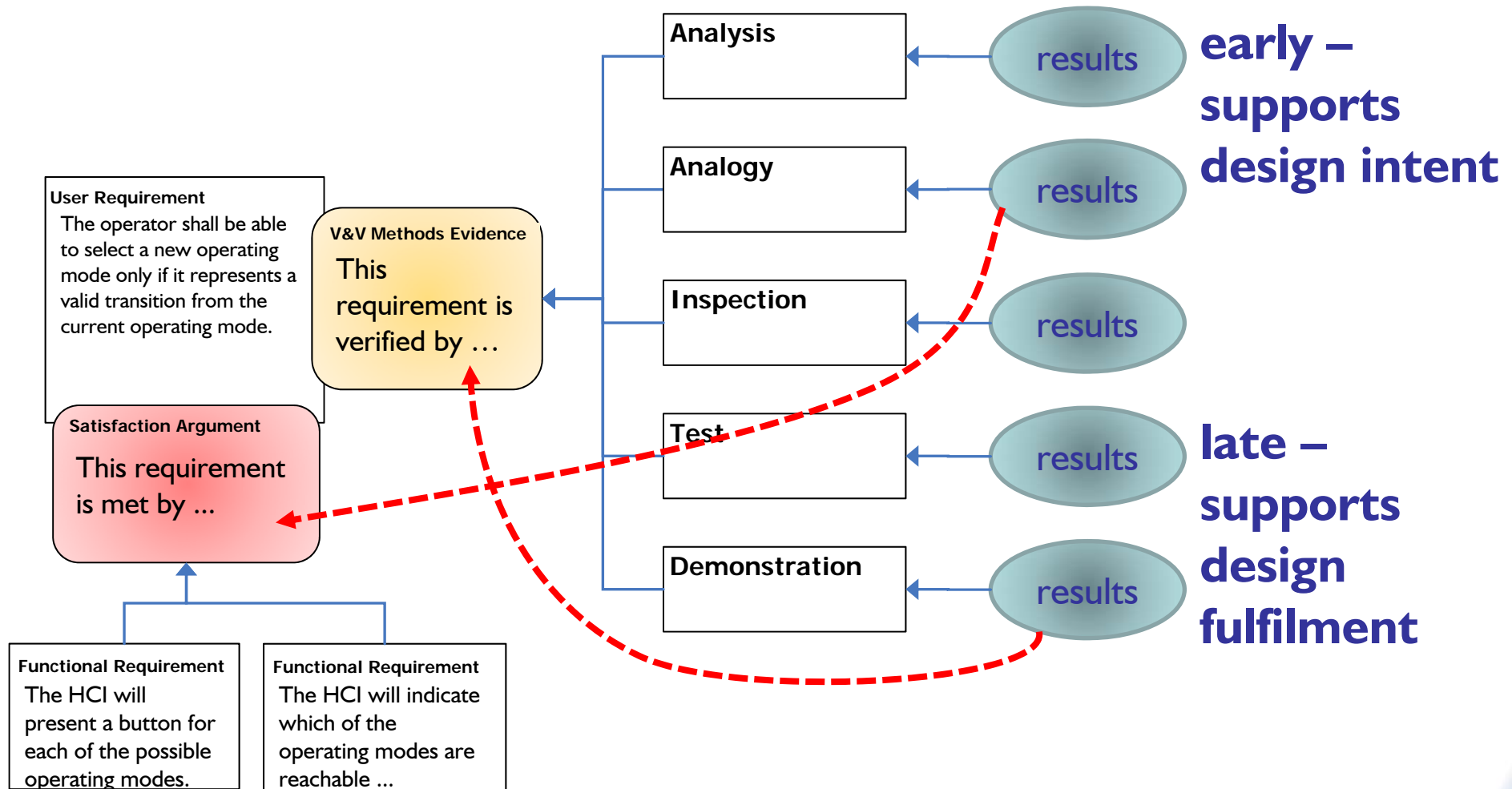
*all of these actions*

***collect evidence***

***for the correctness of the  
design w.r.t. requirements  
(verification)***

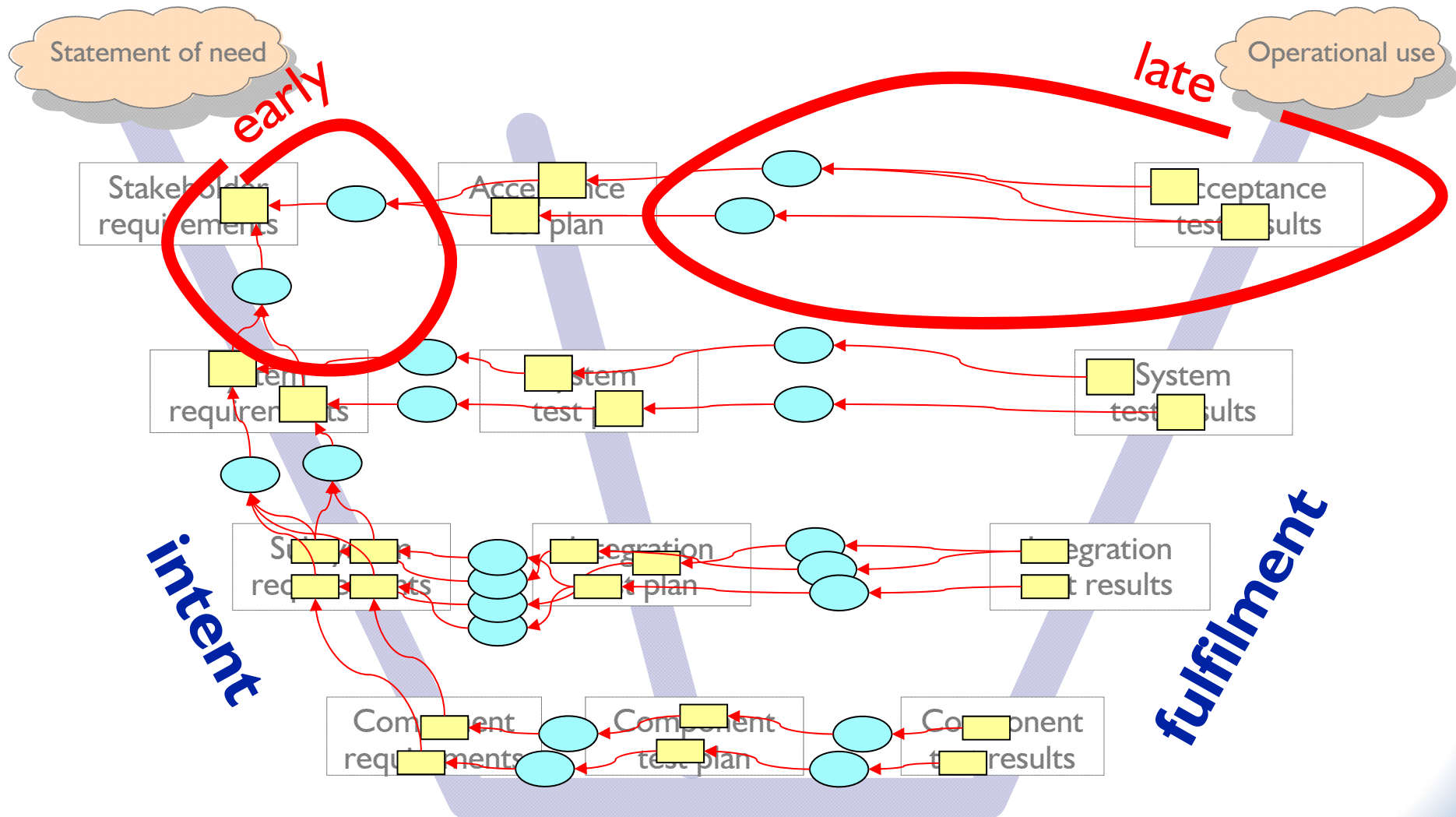


# V&V as “request for evidence”



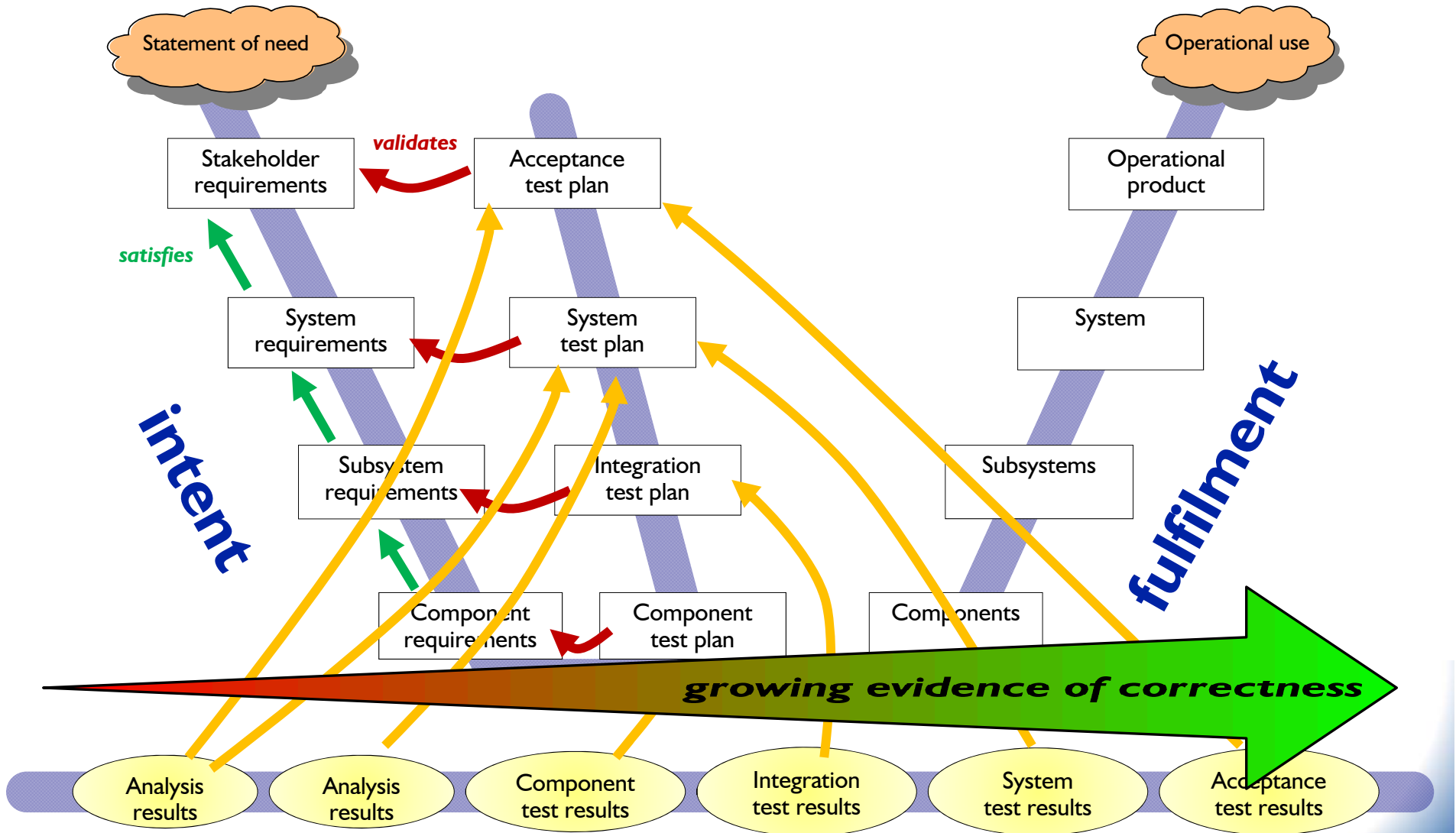


# progressive assurance





# progressive provision of evidence





## summary

- the principles of EbD seem sound
- in practice, a cultural shift is required
  - have to communicate benefits more effectively
  - mentoring in how to write arguments
  - engineers have new emphasis on owning V&V
- effective tool support is vital in visualising data
  - focussed views of local argument structures
  - navigation of large-scale argument structures
- going forward we will:
  - start earlier
  - know better how to write arguments
  - place more emphasis on mentoring



## summary

- using structured arguments within requirements development seems sound
- in practice, a cultural shift is required
  - have to communicate benefits more effectively
  - mentoring in how to write arguments
  - engineers have new emphasis on owning V&V
- effective tool support is vital in visualising data
  - focussed views of local argument structures
  - navigation of large-scale argument structures
- next time we will:
  - start earlier
  - know better how to write arguments
  - place even more emphasis on mentoring