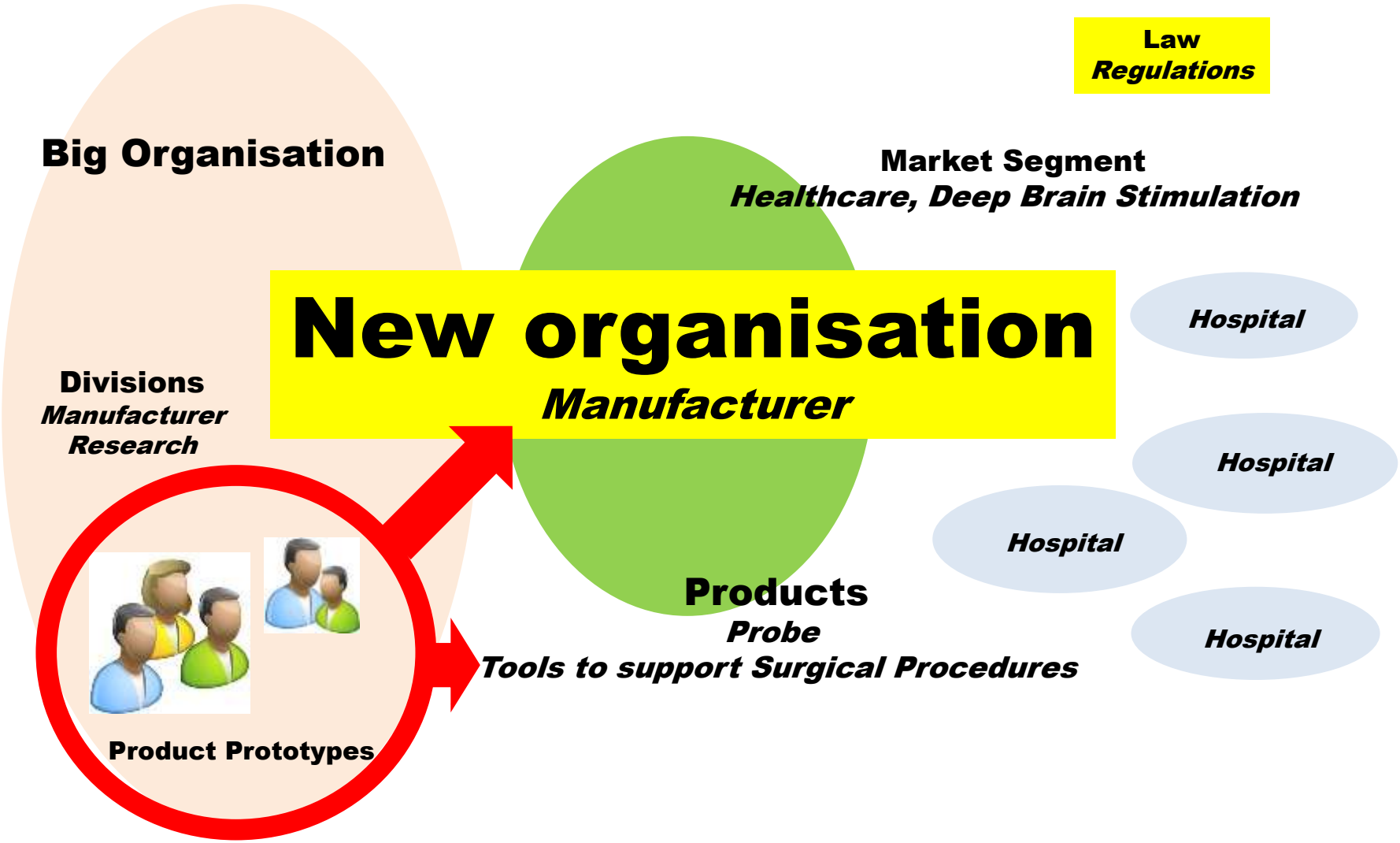
EuroSTAR Software Testing Conference

Testing in a Medical Device Context Limitations are few

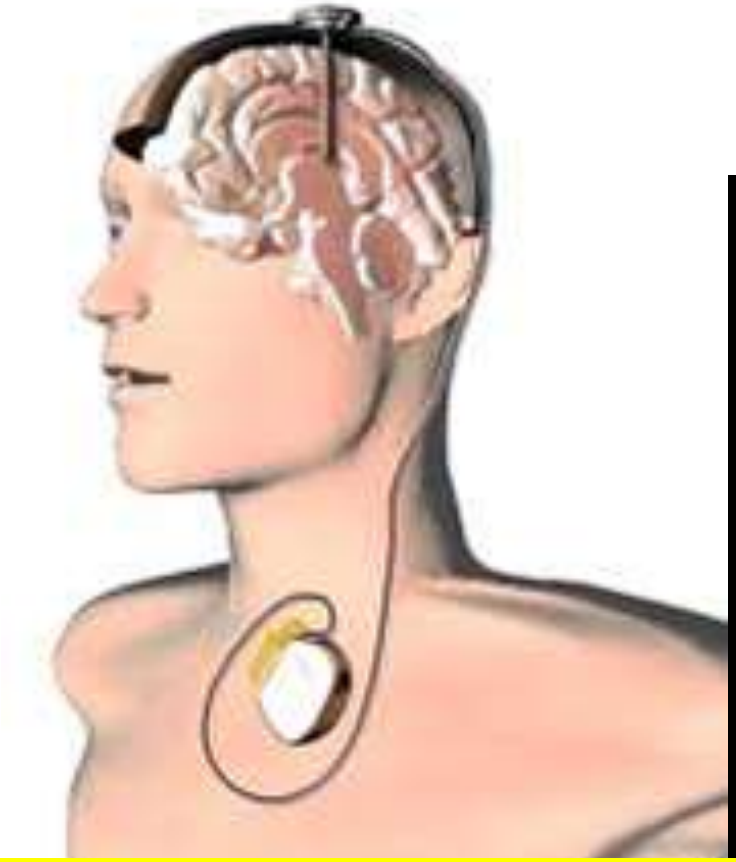
Ruud Cox, Improve Quality Services

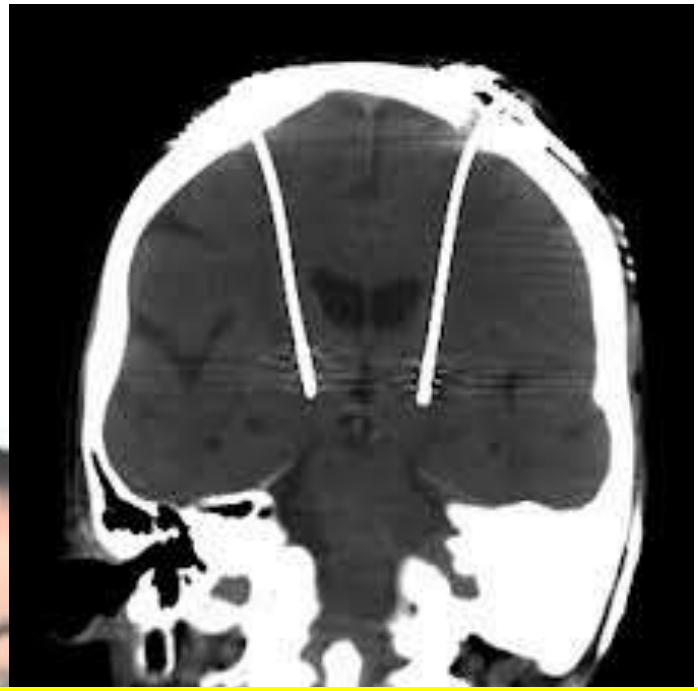
IMPROVE Quality Services

From Research to Manufacturing



Deep Brain Stimulation





A pacemaker for the brain


Almost no testing
Incomplete, overdue documentation

Mission

Organise Testing

(Verification & Validation)

Approach:

Exploratory Testing

Learning, investigation, discovery

followed by

Scripted Testing to demonstrate that the product conforms to specified requirements and to provide objective evidence for submission.

Conformance, Compliance

Exploratory Testing

In the given situation, this was the best way to

- Learn more about the product
- Add value by providing feedback

Charters, Debriefs, Screen Recordings, Managed

A big success

Regulations and Testing

Mission update

Verification only

§820.3(aa) Verification means confirmation by examination and provision of objective evidence that specified requirements have been fulfilled.

Testing is one of many possible verification activities

DESIGN CONTROL GUIDANCE FOR MEDICAL DEVICE MANUFACTURERS
This Guidance relates to FDA 21 CFR 820.30 and Sub-clause 4.4 of ISO 9001

Objective Evidence

Check pressure.

Check that the pressure is OK.

Record the pressure here ____. Check that the pressure is between 50 and 75 psi.

A check requires a decision rule i.e. specific, measurable requirements

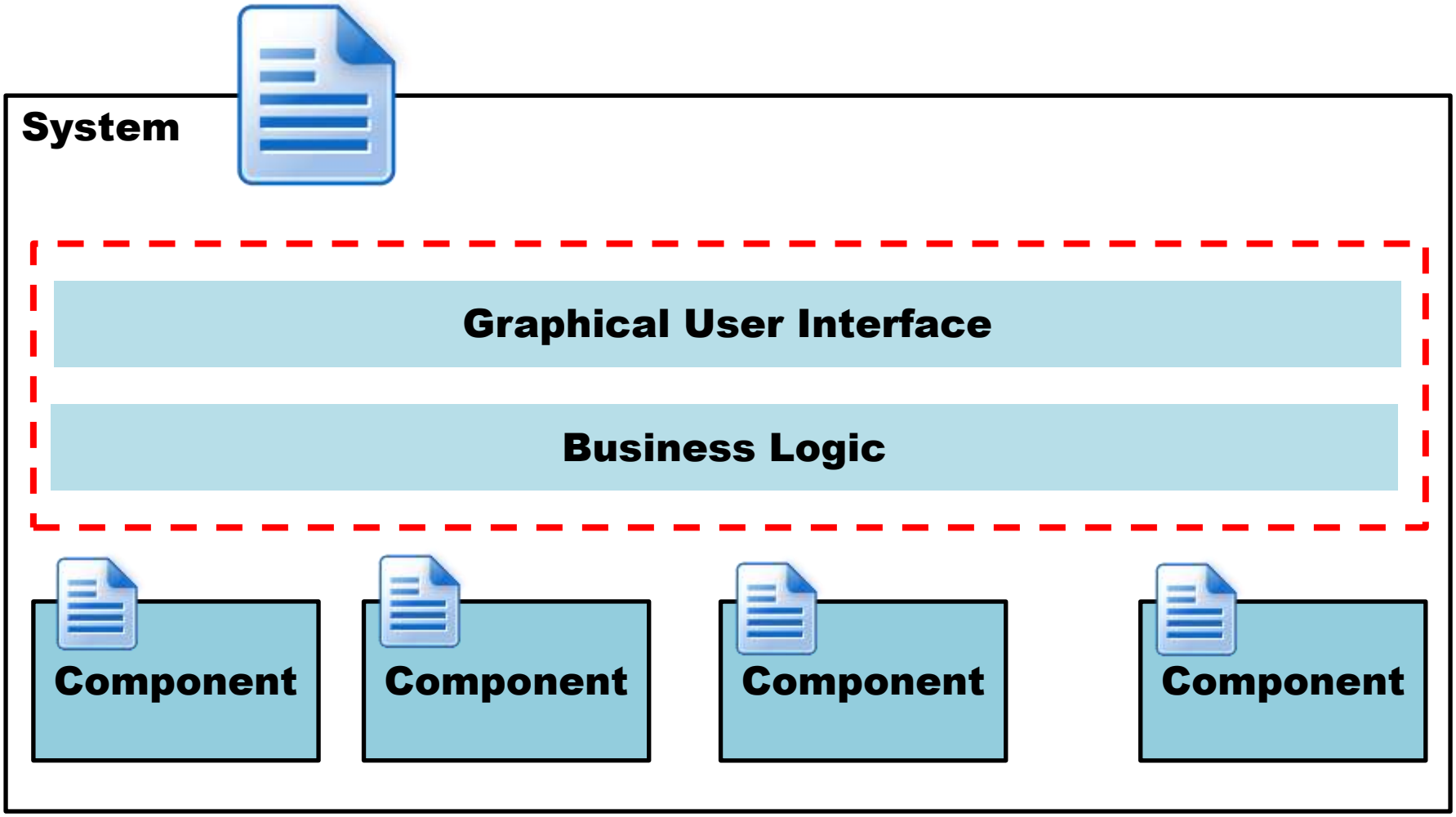
Example from
Medical Device Software Verification, Validation and Compliance by David A. Vogel
ISBN 1596934220

Specified Requirements

- User Stories
  - Sometimes with acceptance criteria
- Traditional requirements
  - "The product shall..."
- User Stories and traditional requirements were not linked

A real challenge!

Global Design



Component Testing

Software Interface

Testability

Exploratory Testing

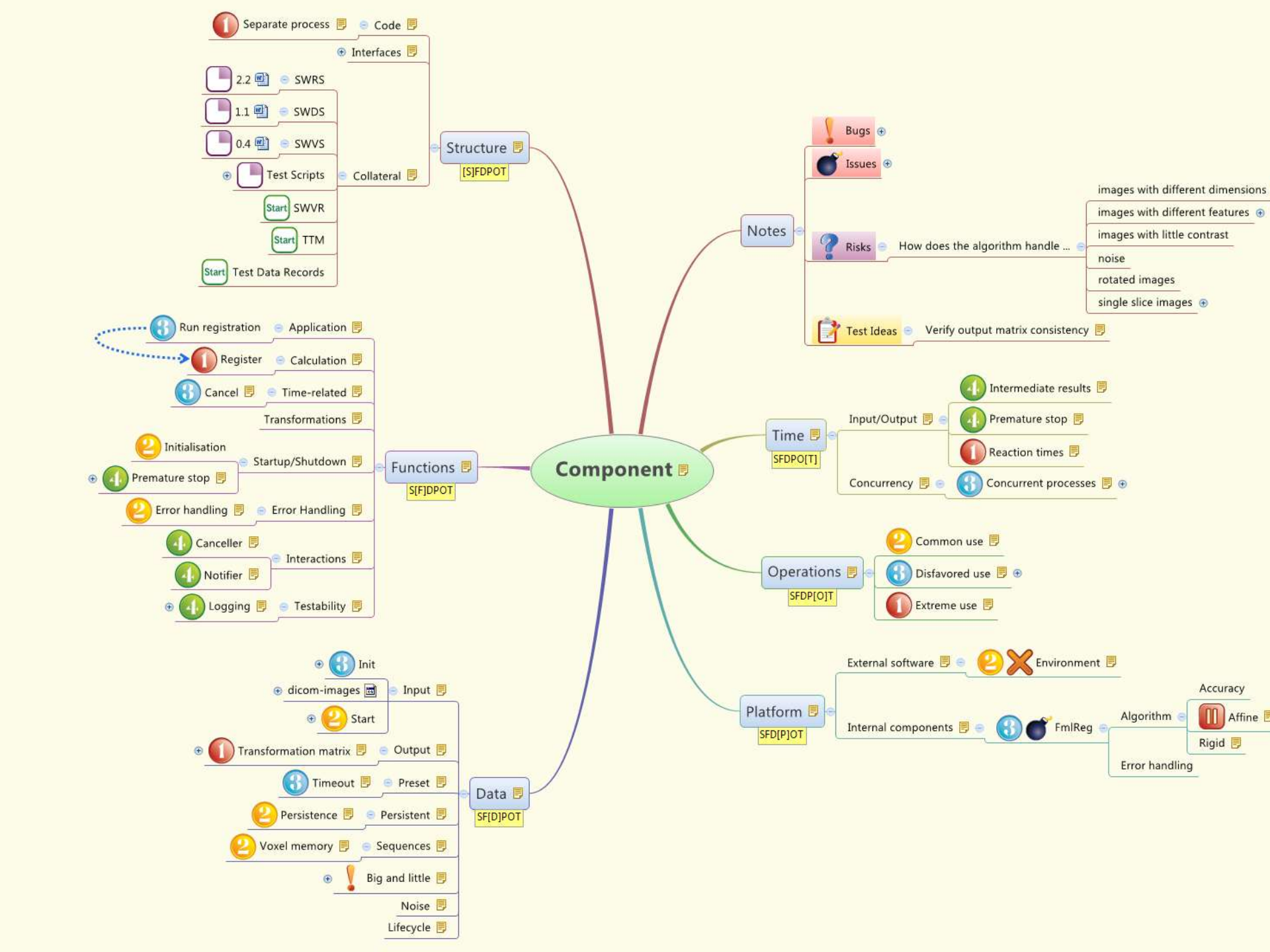
- Coverage/risk outline

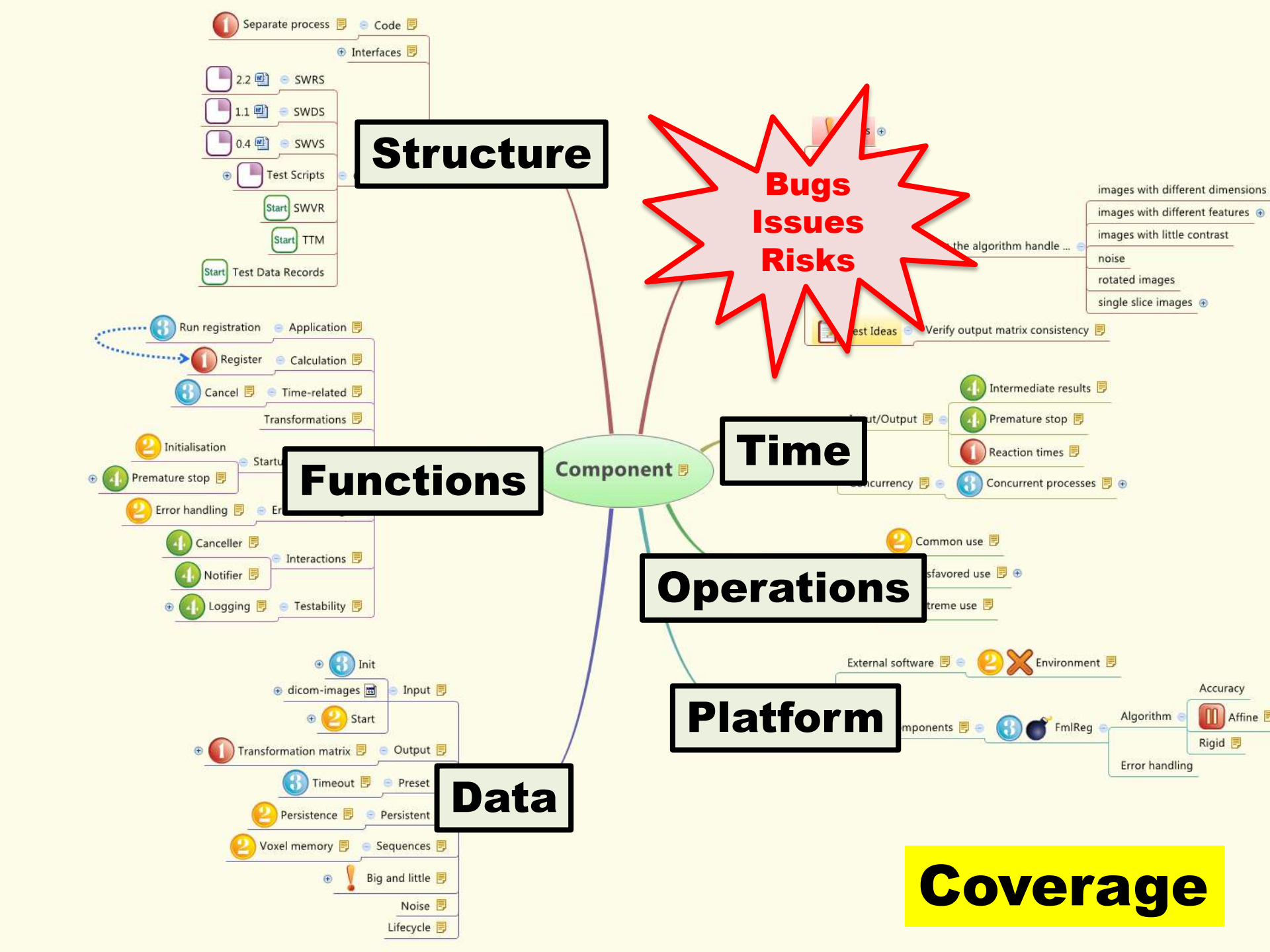
Scripted Testing

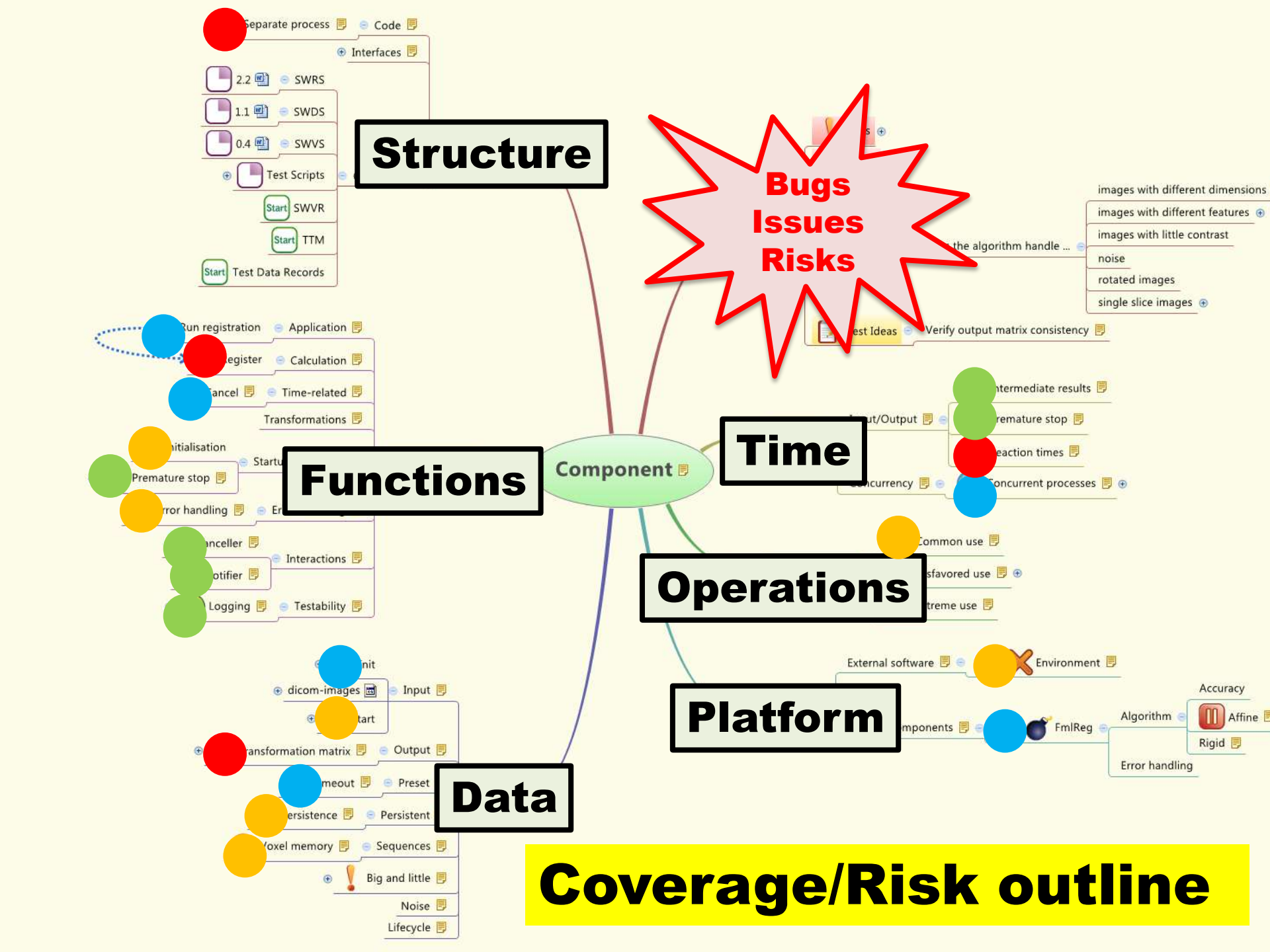
- Unit tests annotated with requirement tags for traceability

```
[Test]
[RequirementTags("Req1", "Req7")]
void FooBar()
{
    Assert.That(…);
}
```

Specify test cases for high risk specified requirements only


Component
Structure
[S]FDPOT
Separate process
Code
Interfaces
2.2
SWRS
1.1
SWDS
0.4
SWVS
Test Scripts
Collateral
Start
SWVR
Start
TTM
Start
Test Data Records
Notes
Bugs
Issues
Risks
How does the algorithm handle ...
images with different dimensions
images with different features
images with little contrast
noise
rotated images
single slice images
Test Ideas
Verify output matrix consistency
Time
SFDPO[T]
Input/Output
Intermediate results
Premature stop
Reaction times
Concurrency
Concurrent processes
Operations
SFDP[O]T
Common use
Disfavored use
Extreme use
Platform
SFD[P]OT
External software
Environment
Internal components
FmlReg
Algorithm
Affine
Rigid
Accuracy
Error handling
Functions
S[F]DPOT
Run registration
Application
Register
Calculation
Cancel
Time-related
Transformations
Initialisation
Startup/Shutdown
Premature stop
Error handling
Error Handling
Canceller
Interactions
Notifier
Logging
Testability
Data
SF[D]POT
Init
dicom-images
Input
Start
Transformation matrix
Output
Timeout
Preset
Persistence
Persistent
Voxel memory
Sequences
Big and little
Noise
Lifecycle


Component
Structure
Separate process
Code
Interfaces
2.2 SWRS
1.1 SWDS
0.4 SWVS
Test Scripts
Start SWVR
Start TTM
Start Test Data Records
Functions
Run registration
Application
Register
Calculation
Cancel
Time-related
Transformations
Initialisation
Premature stop
Error handling
Canceller
Interactions
Notifier
Logging
Testability
Data
Init
dicom-images
Input
Start
Transformation matrix
Output
Timeout
Preset
Persistence
Persistent
Voxel memory
Sequences
Big and little
Noise
Lifecycle
Bugs
Issues
Risks
the algorithm handle ...
images with different dimensions
images with different features
images with little contrast
noise
rotated images
single slice images
Verify output matrix consistency
Time
Intermediate results
Premature stop
Reaction times
Concurrent processes
Operations
Common use
Platform
External software
Environment
FmlReg
Algorithm
Affine
Rigid
Accuracy
Error handling
Coverage


Component
Structure
Separate process
Code
Interfaces
2.2 SWRS
1.1 SWDS
0.4 SWVS
Test Scripts
Start SWVR
Start TTM
Start Test Data Records
Bugs
Issues
Risks
images with different dimensions
images with different features
images with little contrast
noise
rotated images
single slice images
the algorithm handle ...
Verify output matrix consistency
Time
Premature stop
Concurrent processes
Functions
Application
Calculation
Time-related
Transformations
Premature stop
Interactions
Testability
Operations
Common use
Platform
External software
Environment
FmlReg
Algorithm
Affine
Rigid
Accuracy
Error handling
Data
dicom-images
Input
Output
Preset
Persistent
Sequences
Big and little
Noise
Lifecycle
Coverage/Risk outline

Why PASS and FAIL is not enough

PASS only means that no problem was found, which doesn't mean that there's no problem

Comparable Products

- Lots of remarks during evaluation at hospital
- Our product side-by-side to competitors product

Independent Testing

- Testers were reponsible for
  - ALL test documentation
  - Providing Objective Evidence
- Developers did some unit testing but
  - Their tests were undocumented
  - Could not be used as Objective Evidence

Why does testing take so long?

JIGGLE!

Time-out

Testing

Incomplete, overdue specified requirements. Independent testing. “I don’t know nothing about testing.” What does the team **know** about or does have **experience** with working in a **Medical Device Context?**

Agile/Scrum

Almost no Agile practices. The team didn’t live Scrum. What does the team **know** about or does have **experience** with **Agile/Scrum?**

Regulations and Agile/Scrum

CONCURRENT ENGINEERING. Although the waterfall model is a useful tool for introducing design controls, its usefulness in practice is limited.

DESIGN CONTROL GUIDANCE FOR MEDICAL DEVICE MANUFACTURERS
This Guidance relates to FDA 21 CFR 820.30 and Sub-clause 4.4 of ISO 9001

Agile TIR SW Committee Draft 1.0

Guidance on the use of AGILE practices in the development of medical device software

Improvements after Time-out

Testing

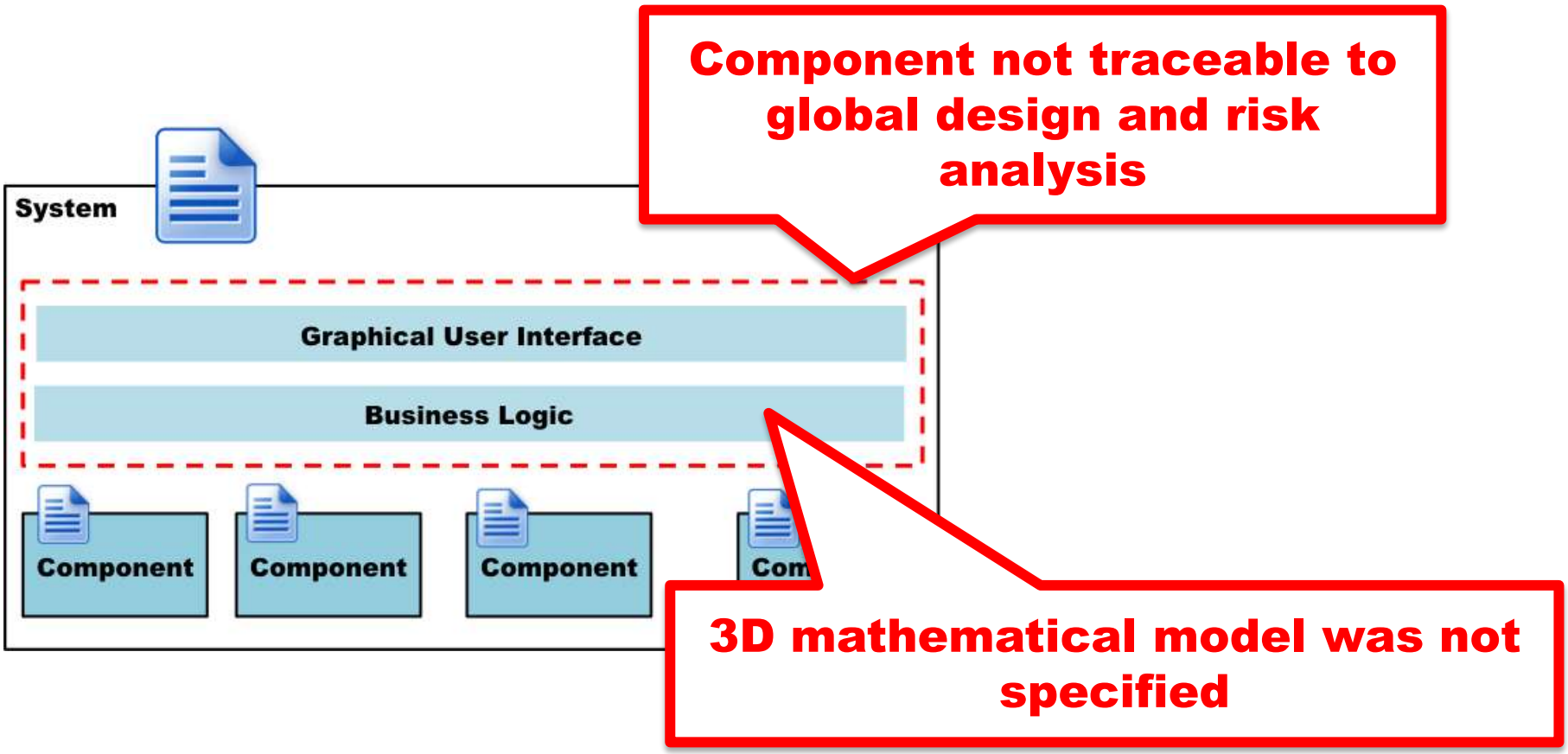
- IEC 62304 training, Less independent testing.

Agile/Scrum

- Product Backlog, Sprint Planning, Definition of Done, Scrum Board, Sprint Review, Retrospectives, Smaller Teams each with their own Scrum Master
- Scrum Master, Product Owner training
- Agile/Scrum Coach

Overdue specified requirements

Other Identified Risks



Claim these results

Product Owner

System Architect

Clarify mission

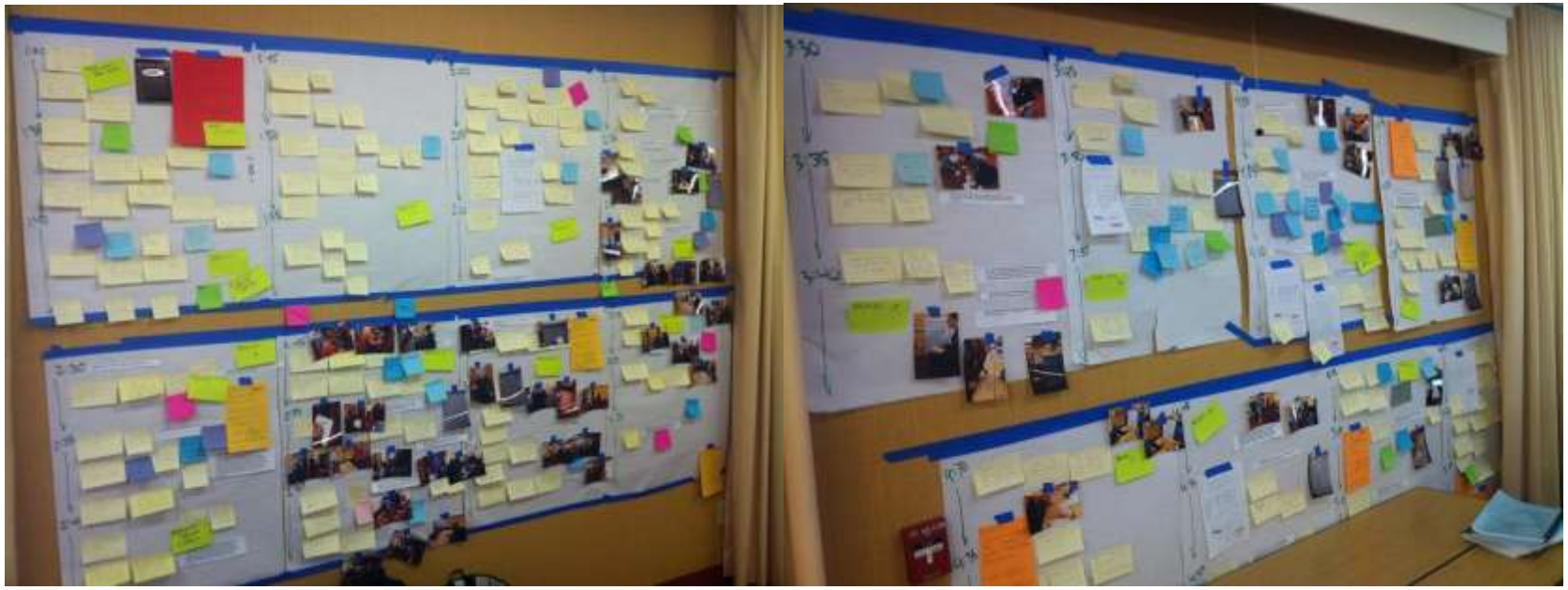
Program Director

Test Architect

Director
Quality & Regulations

Status reporting

Observe the work of testing



A diary might help

Lessons Learned, Conclusions

- A feedback loop is mandatory
  - Mission
  - Status Reporting
- Medical Device Context is not limiting
  - Exploratory testing is possible
  - Agile/Scrum is possible

No matter what the problem is, it's always a people problem.

Gerald M. Weinberg

Questions

Ruud Cox

email

rco@improveqs.nl

Twitter

@ruudcox

Blog

http://ruudcox.wordpress.com/