verum®
software by design

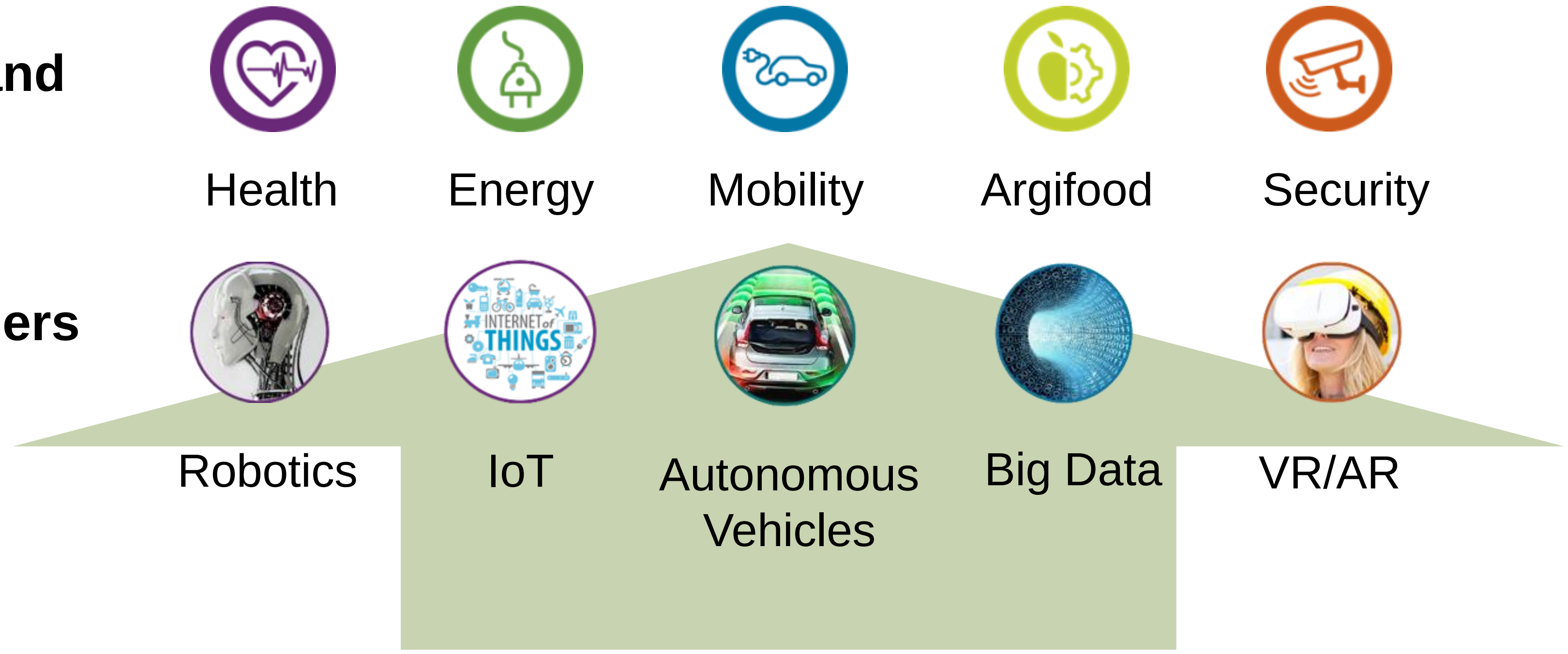
Discover Dezyne

The easiest way to build verifiably correct embedded software

Component based software testing

May 2017

Why is software relevant?



Embedded Software

Driving Emerging Solutions

Why be concerned about software?

Complexity
Reliability
Connectivity
Security
Safety
Cost
Flexibility
Robustness

Why Verum?

- We stand for more efficient, effective and economic ways of building and testing high-tech software systems
- Our product, Dezyne, enables engineers to specify, design, validate and formally verify software components for embedded, industrial & technical software systems
- It delivers a range of business benefits:

Factor 2-3 increase in efficiency

20% decrease in time to market

160x decrease in field defects

Where is Dezyne used?

verum®
software by design

Introducing Dezyne

dezyne™
create explore verify trust

DEZYNE modeling language
specifications and designs of software systems

DEZYNE simulation
understand & validate specifications & designs

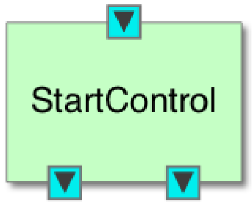
DEZYNE verification
prove that specifications & designs are complete and correct

DEZYNE code generation
generate efficient, reliable and robust code

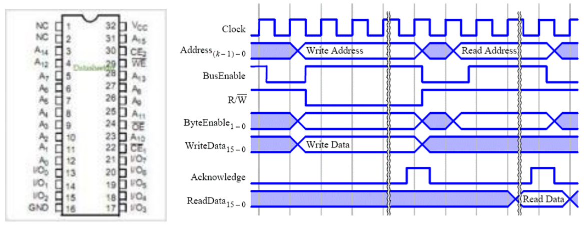
Component based design

Hardware Component

Dezyne Component

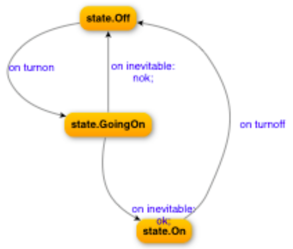


Component Specification

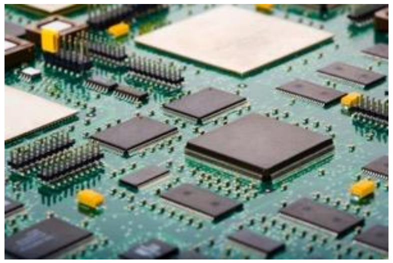


Interface Specification

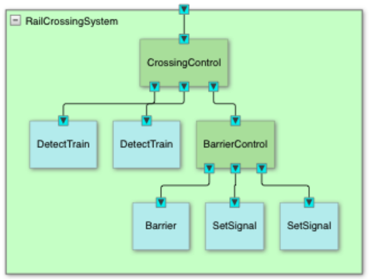
```
in void turnon( in initData iD1 );
out void ok();
```



System Model

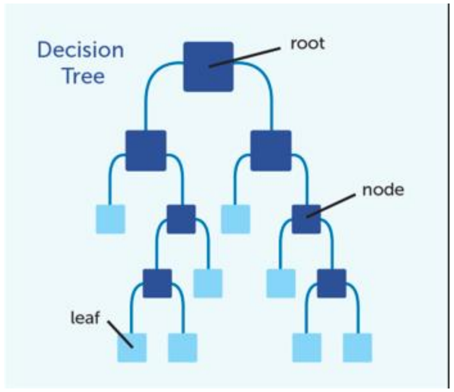


System Model



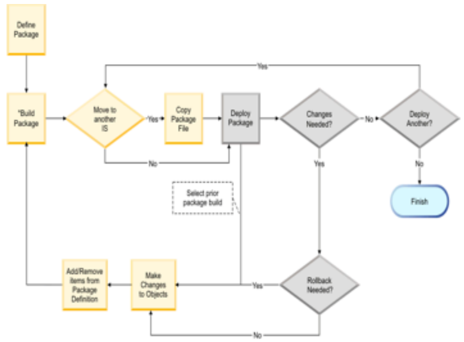
Applications of Dezyne

Decision Tree

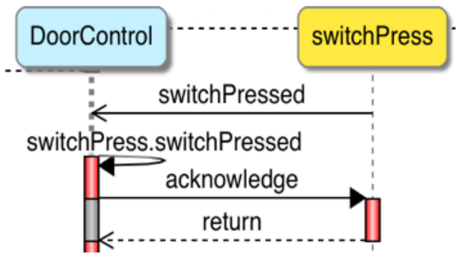


Sequencing/ Cyclic Control

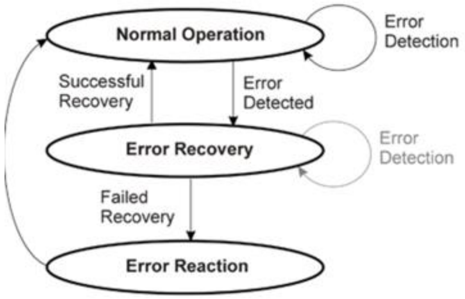
System Workflow



Event Handling

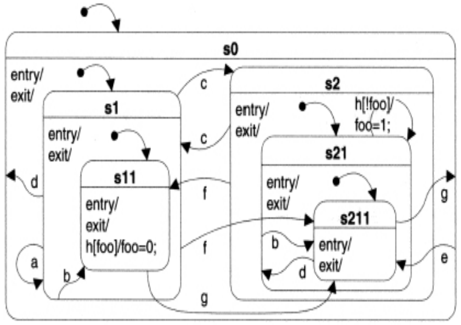


Error Handling & Recovery

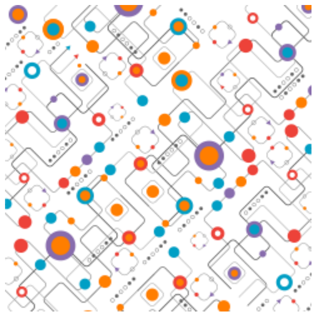


Logical Relationships

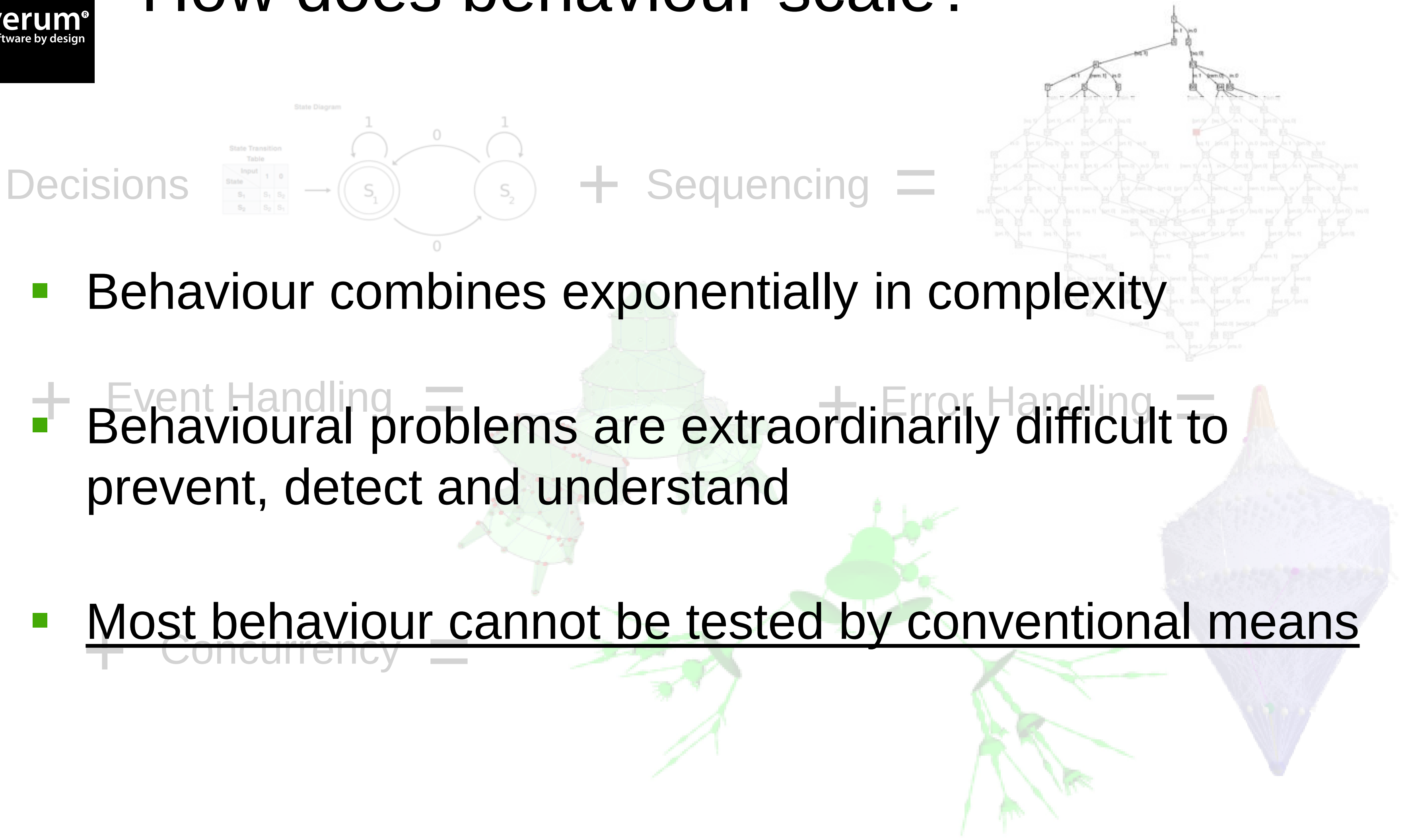
Concurrency



Communication & Routing

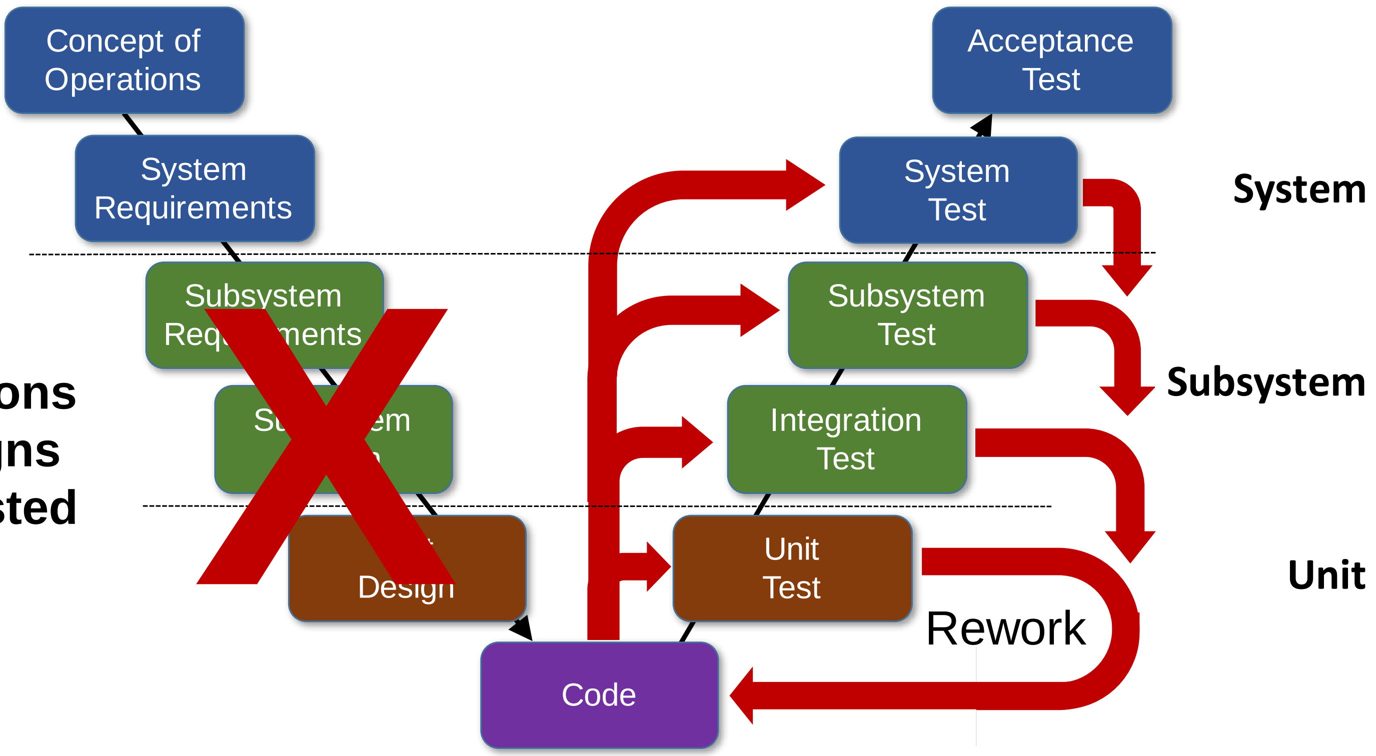


How does behaviour scale?



- Behaviour combines exponentially in complexity
- Behavioural problems are extraordinarily difficult to prevent, detect and understand
- Most behaviour cannot be tested by conventional means

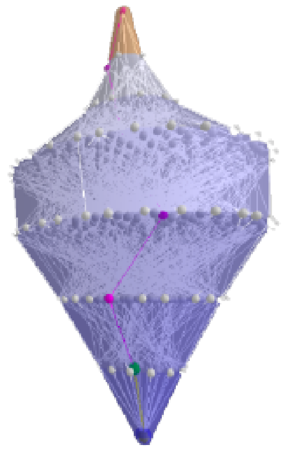
How conventional development works



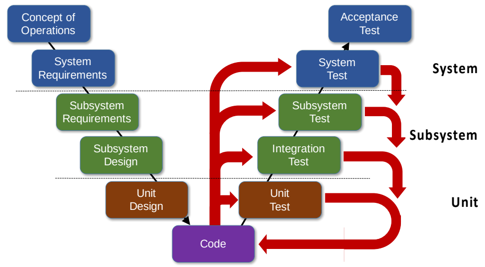
Specifications and designs are not tested

Conventional software testing starts with code
The result is rework, late in the lifecycle

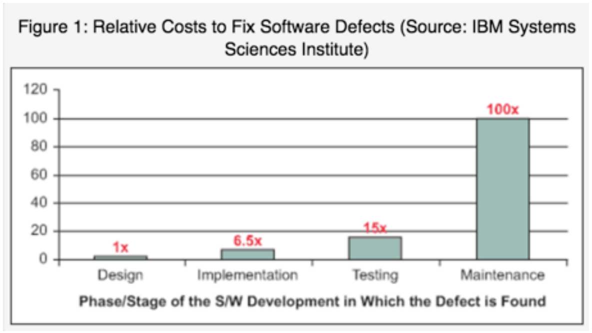
What are the consequences?



+



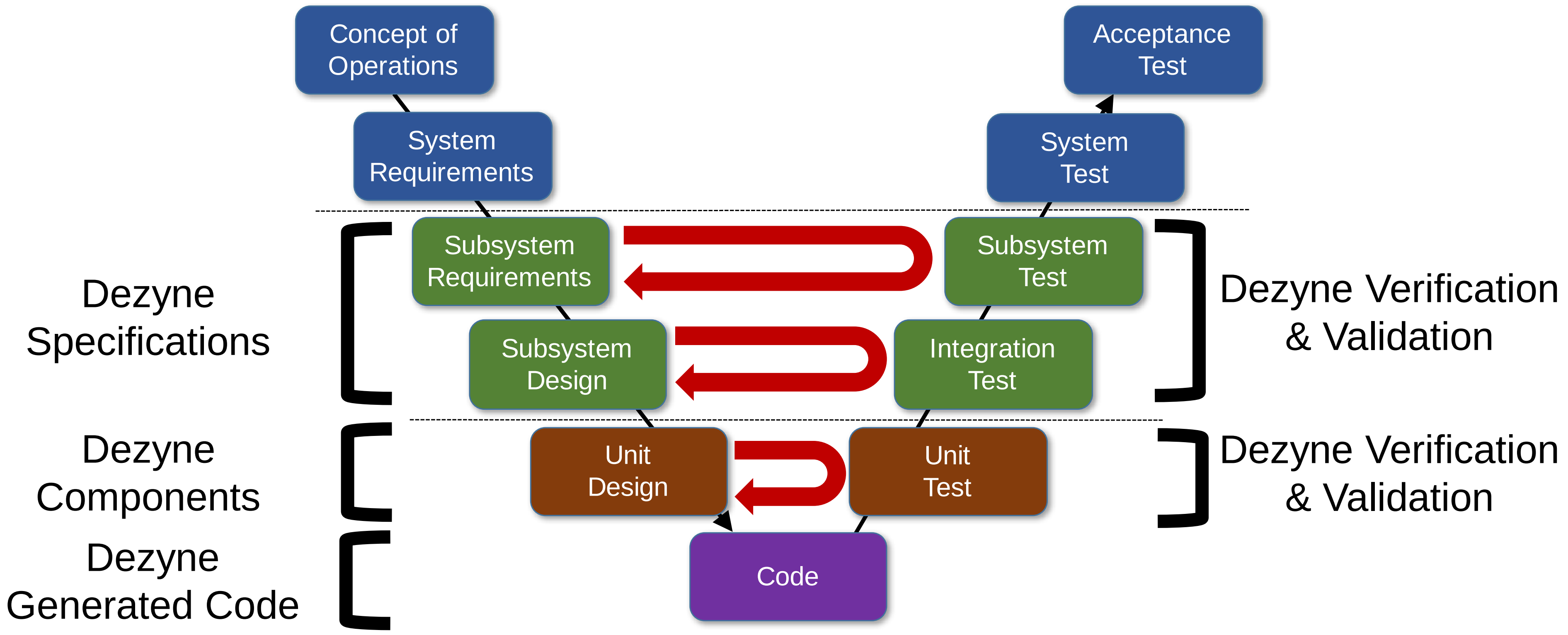
+



=

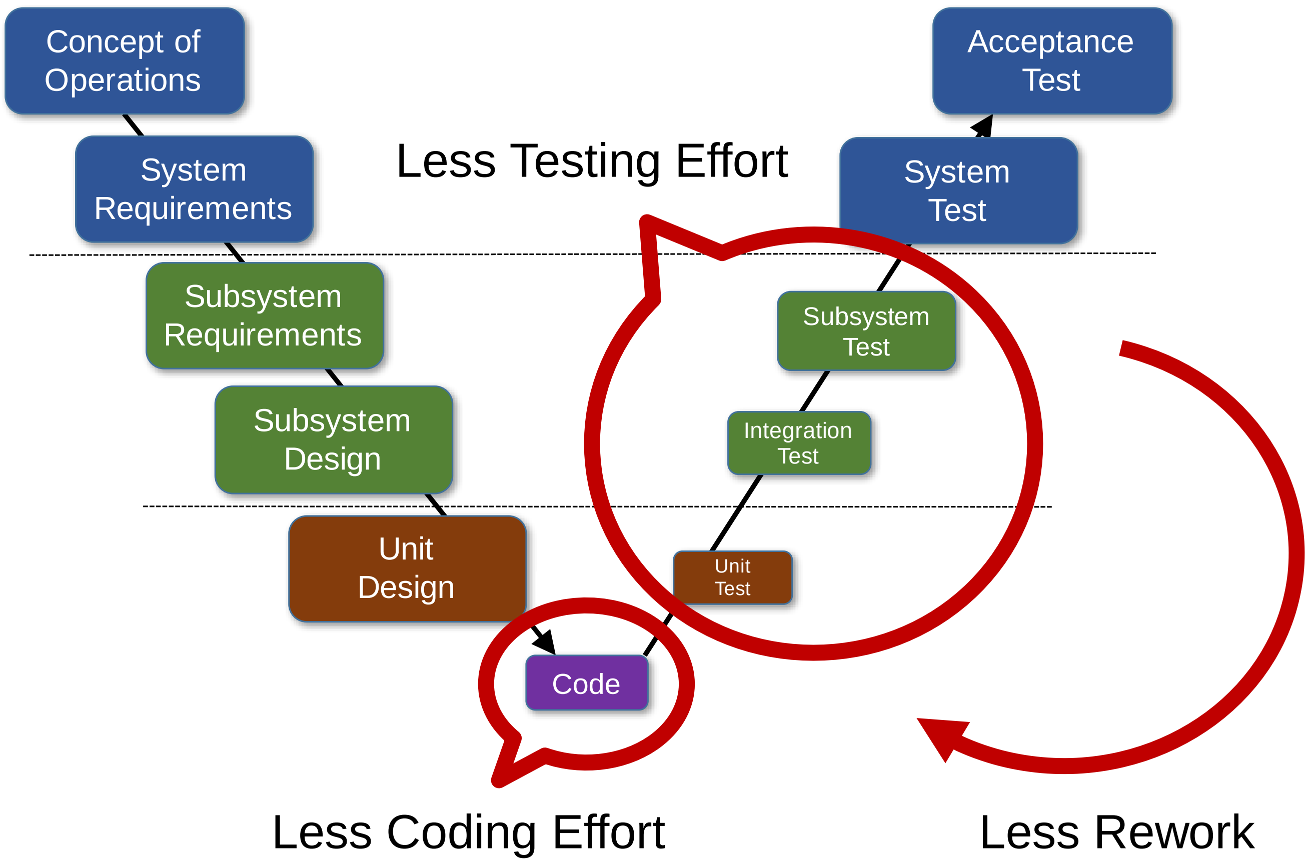
- Poor Predictability
- High Costs
- Delays
- Questionable Quality

Defect prevention is better than cure

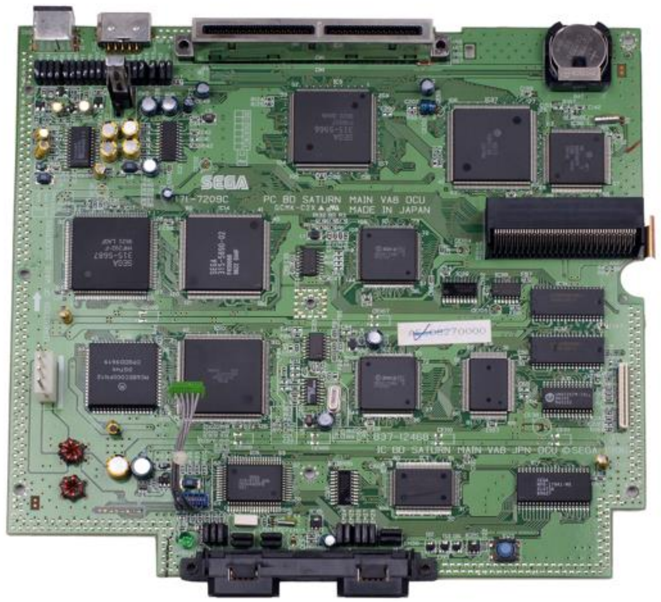


Dezyne Specifications & Components can be continuously, automatically tested

Preventing defects saves €€€



Why components?



Each component
has an explicit
interface specification
Data is separated
from Control

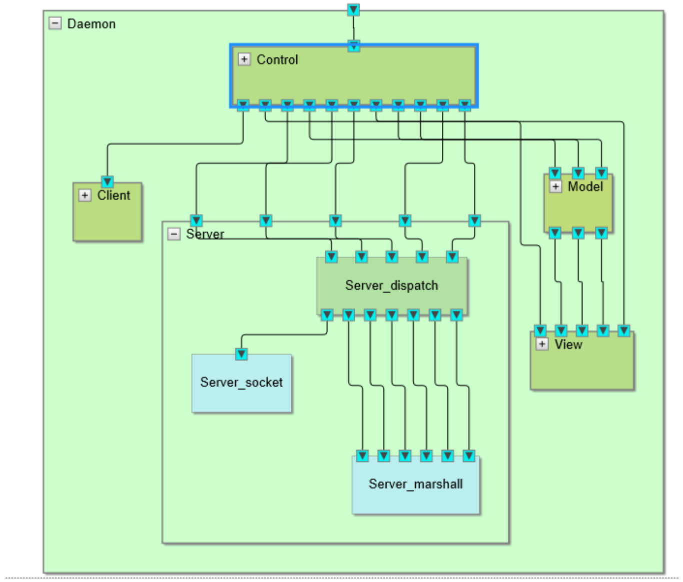
The functionality of
each component
is distinct and serves
a specific purpose

Low Coupling

High Cohesion

Maintainable, Extensible and Reusable Hardware

Why components?



Each component has an explicit interface specification
Data is separated from Control

The functionality of each component is distinct and serves a specific purpose

Low Coupling

High Cohesion

Maintainable, Extensible and Reusable Software

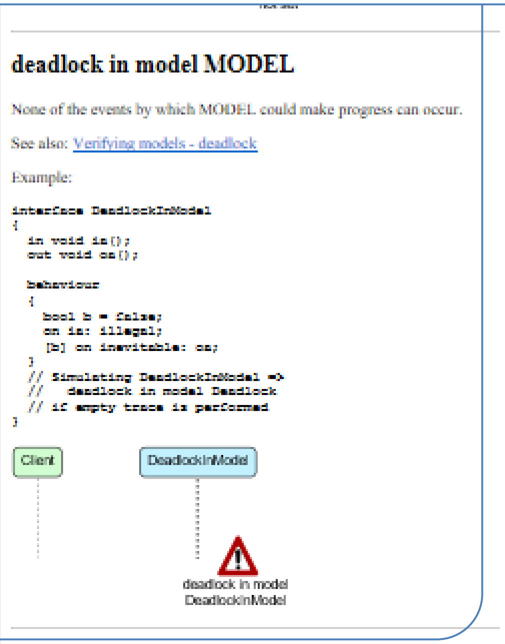
“Separation of Concerns”

Always up to date documentation

Input specification

- •Free text with formatting
- •Model(s)
- •Test scenarios / traces
- •Dezyne Diagrams
- •Test Sequence output
- •Verifier report statistics
- •Regression tests

HTML output



deadlock in model MODEL

None of the events by which MODEL could make progress can occur.

See also: Verifying models - deadlock

Example:

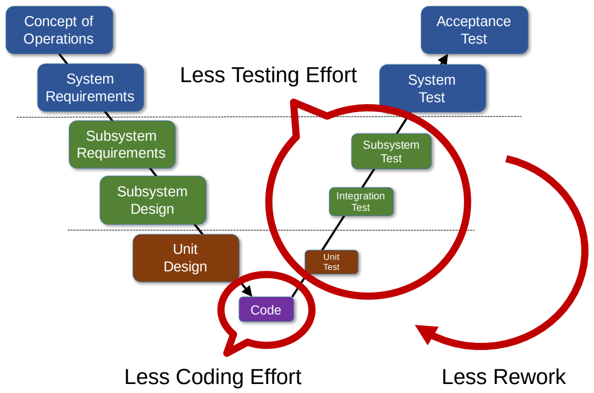
```
interface DeadlockInModel
{
  in void ia();
  out void oa();

  behaviour
  {
    bool b = false;
    on ia: illegal;
     [b] on inevitable: oa;
  }
  // Simulating DeadlockInModel =>
  //    deadlock in model Deadlock
  // if empty trace is performed
}
```

Documentation is generated directly from component specifications and designs

Summary: benefits of MDE

Economically realising reliable, robust software

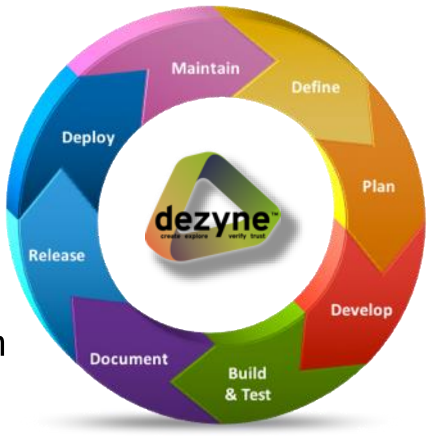


Reducing product lifecycle management costs

Lower maintenance costs

Lower deployment costs

Lower documentation costs



Lower design costs

Lower development costs

Lower testing costs

verum®
software by design

Discover Dezyne

The easiest way to build verifiably correct embedded software

Thanks for your attention