No Test without Integration

October 6th, 2015 – Dutch Testing Day

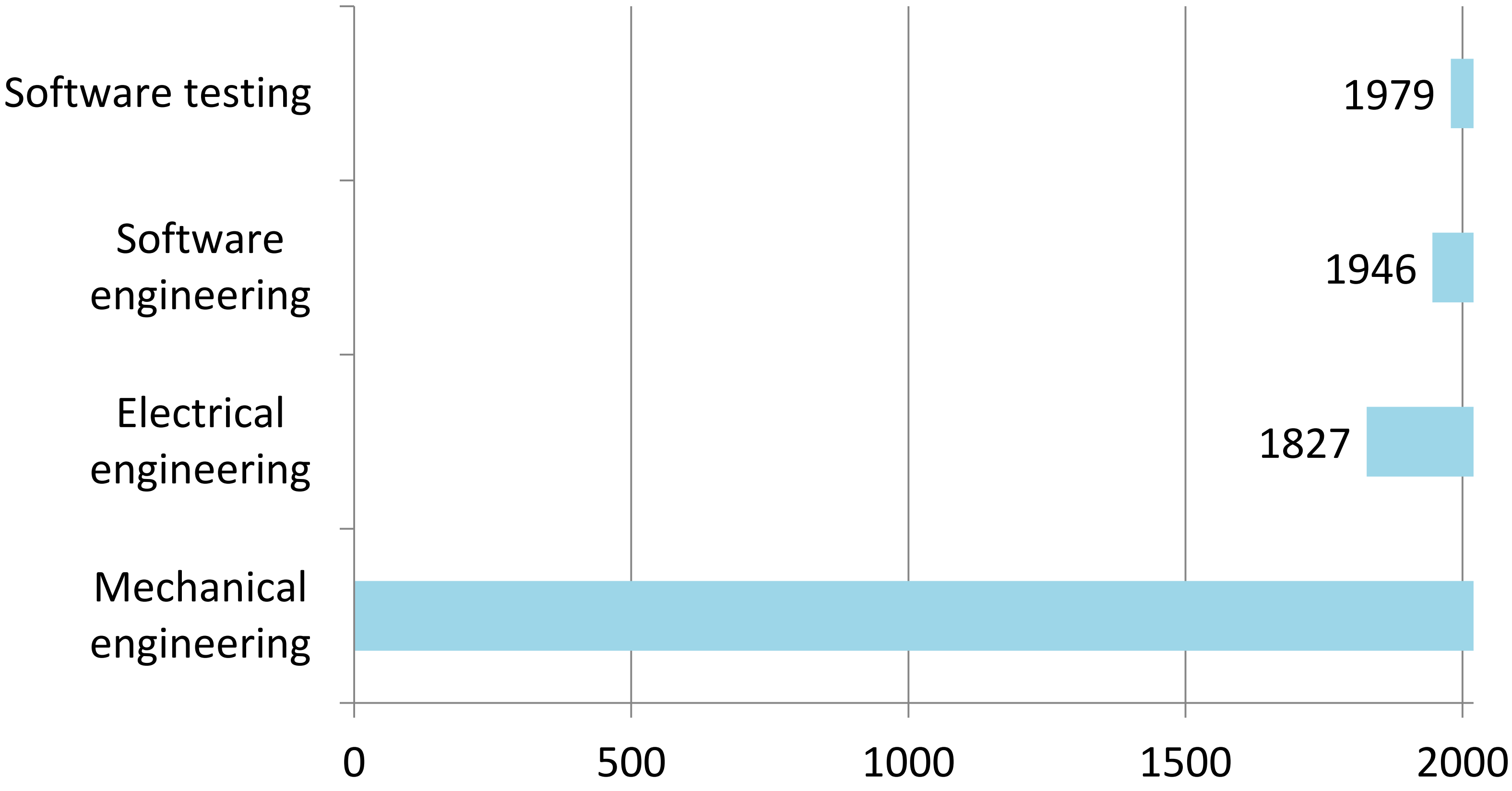
André Verschelling

Sr. Test Consultant

ALTEN Nederland

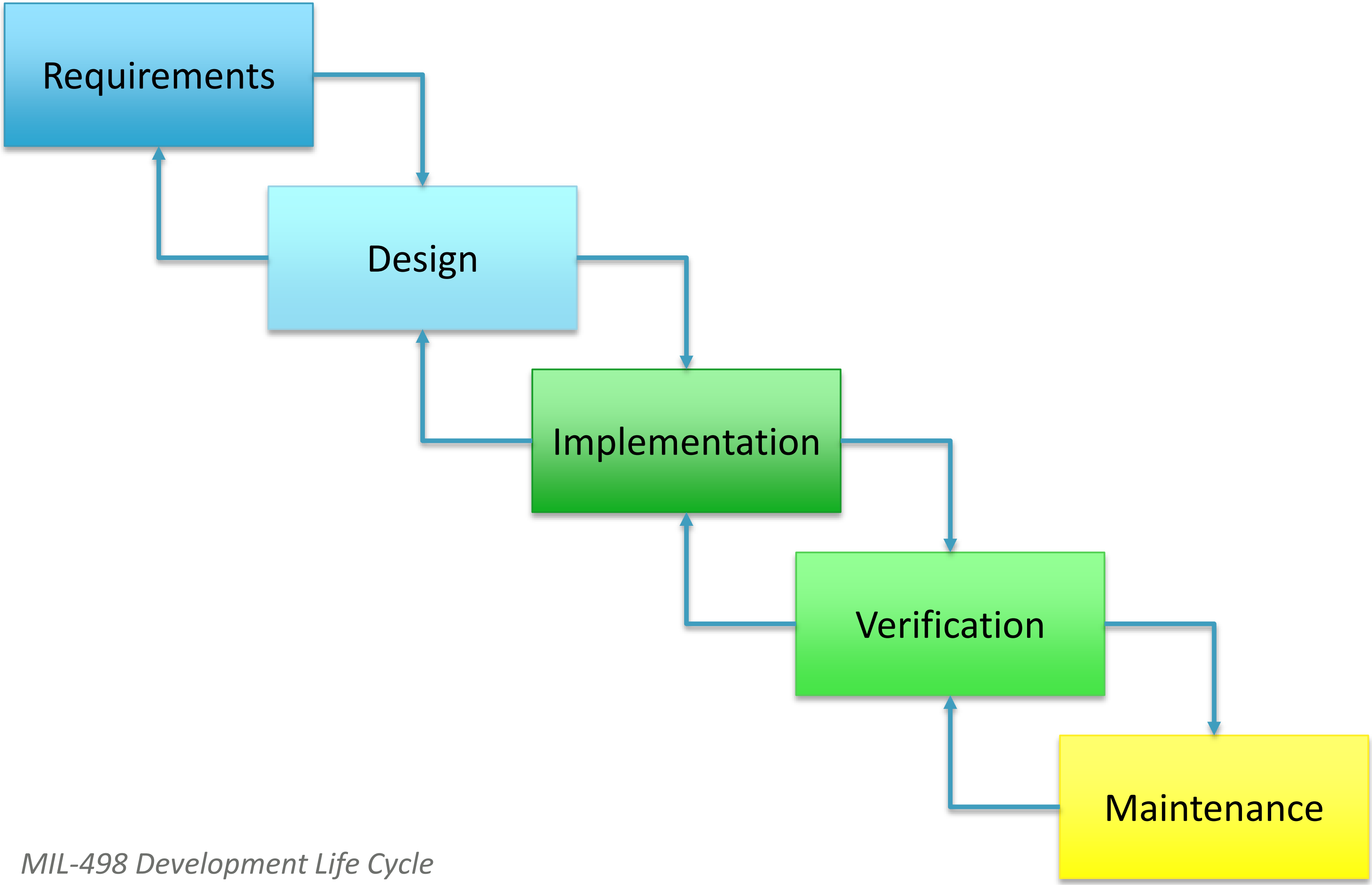
Our testing challenge

Our testing knowledge



A young discipline still, but ever since it started we have been improving and optimizing

We changed sequential processes …



Requirements

Design

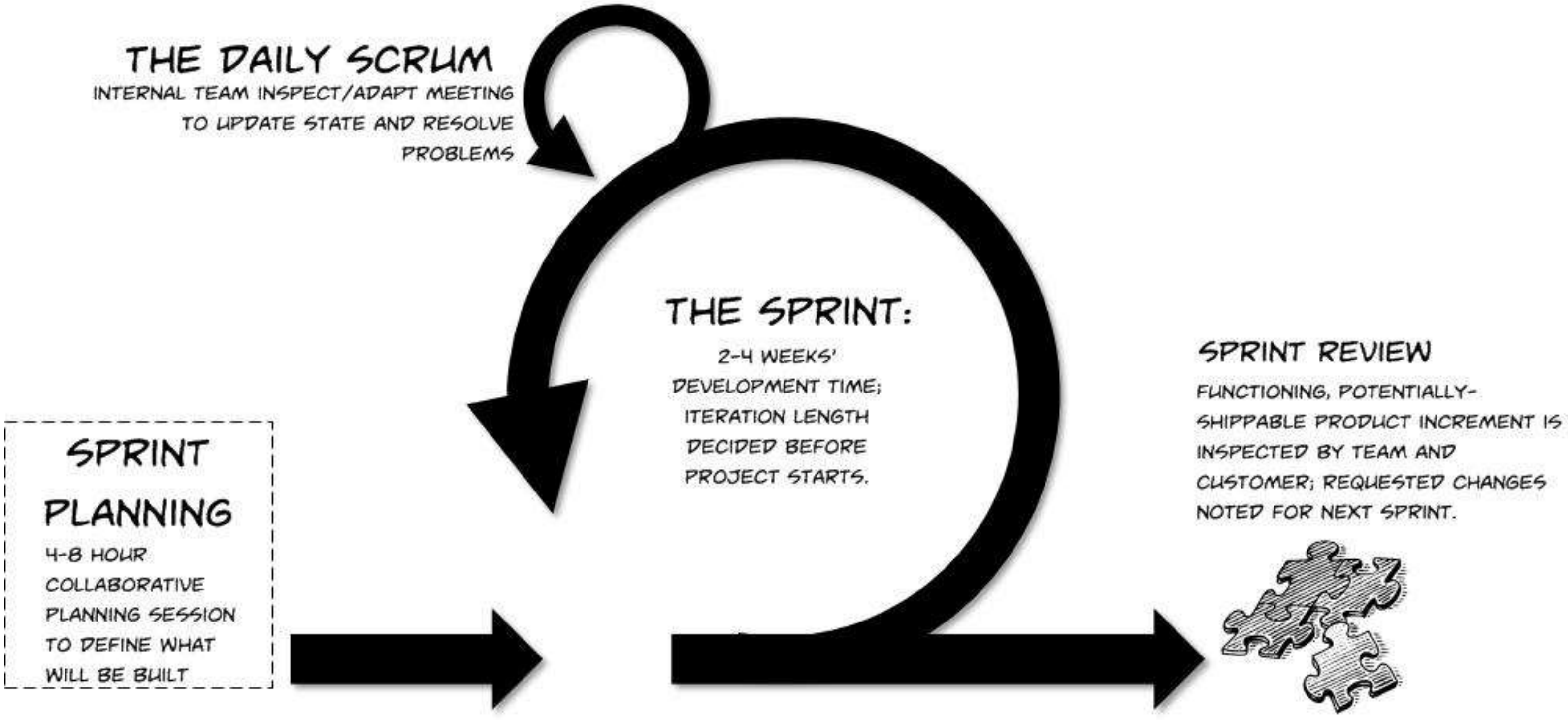
Implementation

Verification

Maintenance

MIL-498 Development Life Cycle

… into Agile methodologies



THE DAILY SCRUM

INTERNAL TEAM INSPECT/ADAPT MEETING TO UPDATE STATE AND RESOLVE PROBLEMS

SPRINT PLANNING

4-8 HOUR COLLABORATIVE PLANNING SESSION TO DEFINE WHAT WILL BE BUILT

THE SPRINT:

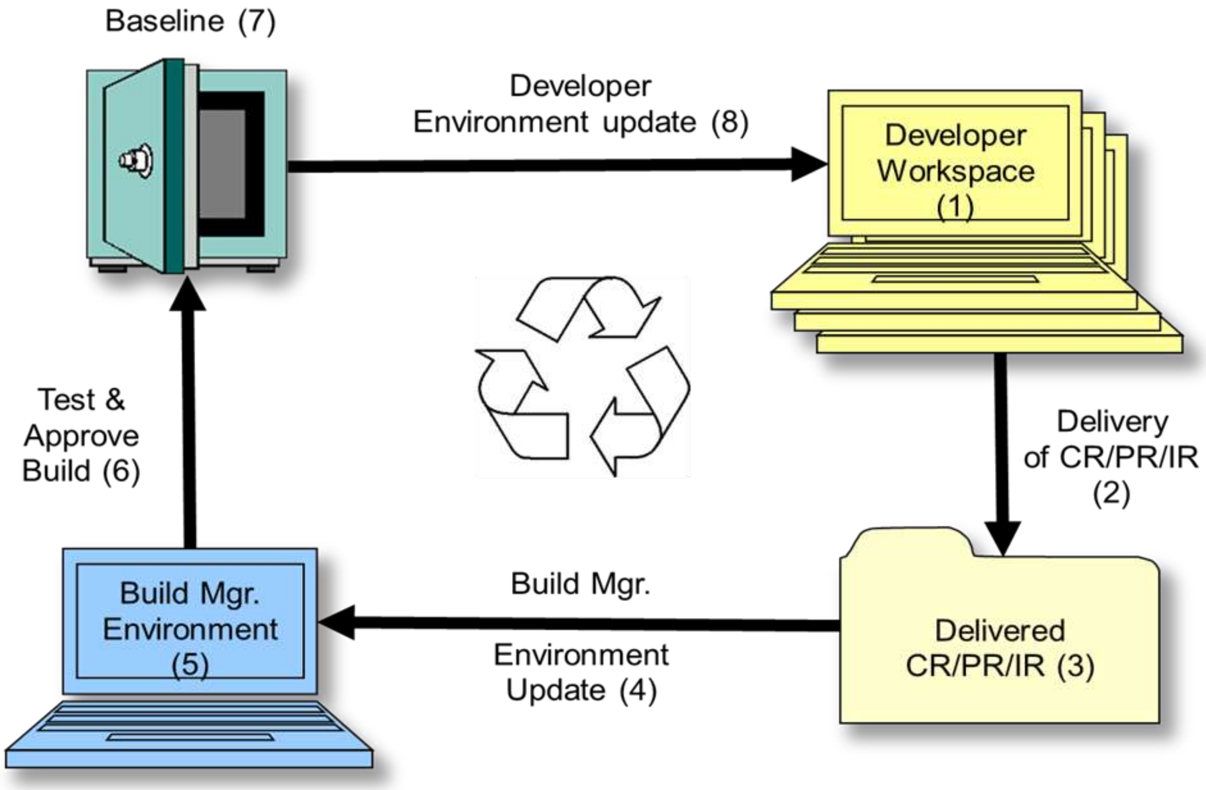
2-4 WEEKS' DEVELOPMENT TIME; ITERATION LENGTH DECIDED BEFORE PROJECT STARTS.

SPRINT REVIEW

FUNCTIONING, POTENTIALLY-SHIPPABLE PRODUCT INCREMENT IS INSPECTED BY TEAM AND CUSTOMER; REQUESTED CHANGES NOTED FOR NEXT SPRINT.

http://www.scrumhint.com/wp-content/uploads/2014/04/canada.jpg

And build manager controlled …



Baseline (7)

Developer Environment update (8)

Developer Workspace (1)

Delivery of CR/PR/IR (2)

Delivered CR/PR/IR (3)

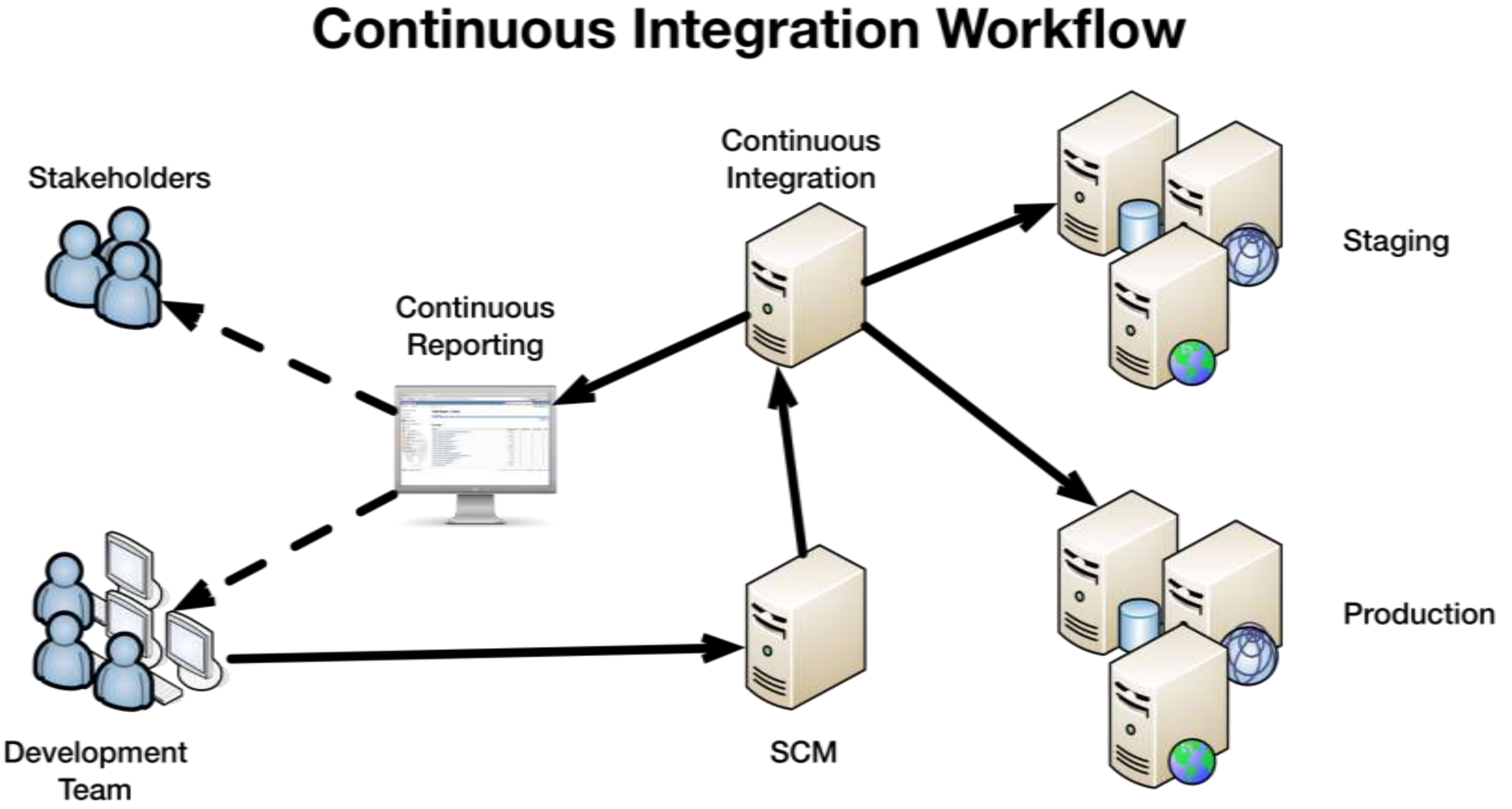
Build Mgr. Environment Update (4)

Build Mgr. Environment (5)

Test & Approve Build (6)

Example configuration Philips Research

… to autonomous builds


Continuous Integration Workflow
Stakeholders
Continuous Integration
Staging
Continuous Reporting
Production
Development Team
SCM

http://www.in2it.be/wp-content/uploads/2013/12/Continuous-Integration-Workflow.png

Can we optimize any further?

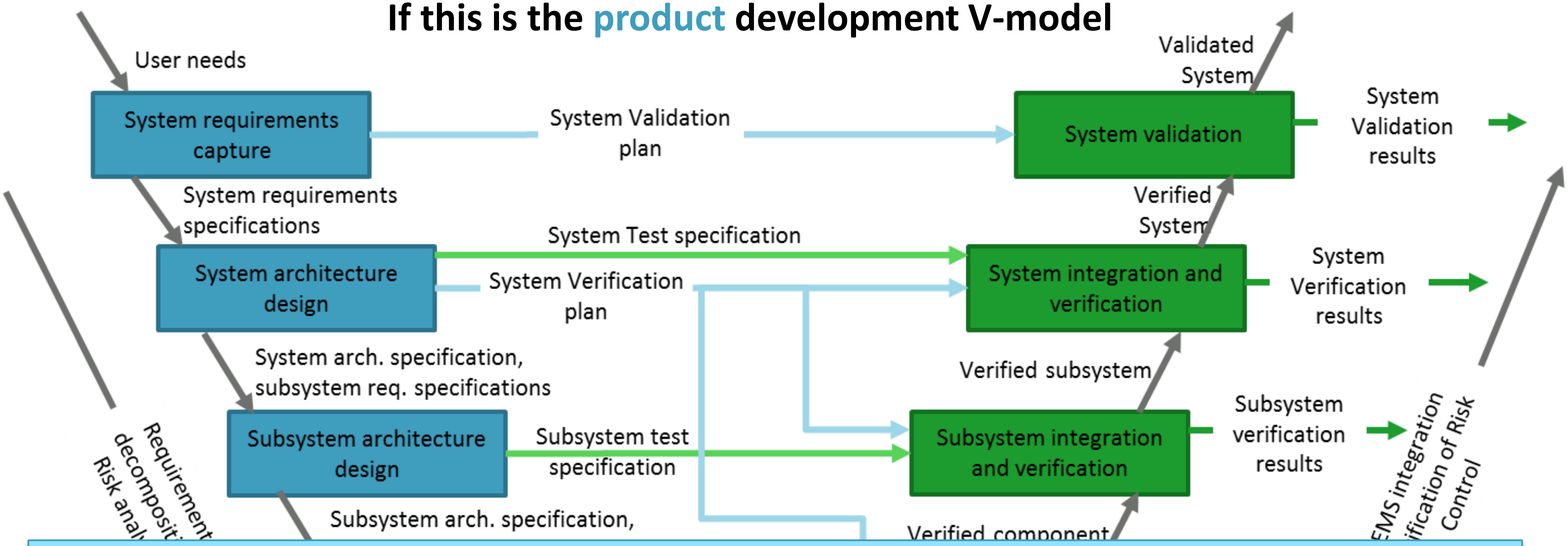
Integration without Test

Continuous Integration in Agile software development results in early defect detection without noticeable test effort. Boundaries between testing and developing are fading. **Is it possible to reinvent system integration in such a way, that testing will become superfluous?**

  - *Call for Proposals Dutch Testing Day 2015*

Where did we optimize?

If this is the product development V-model



User needs

System requirements capture

System Validation plan

System validation

Validated System

System Validation results

System requirements specifications

System Test specification

Verified System

System architecture design

System Verification plan

System integration and verification

System Verification results

System arch. specification, subsystem req. specifications

Verified subsystem

Subsystem architecture design

Subsystem test specification

Subsystem integration and verification

Subsystem verification results

Subsystem arch. specification,

Verified component

Requirement

decomposit

Risk anal

EMS integration

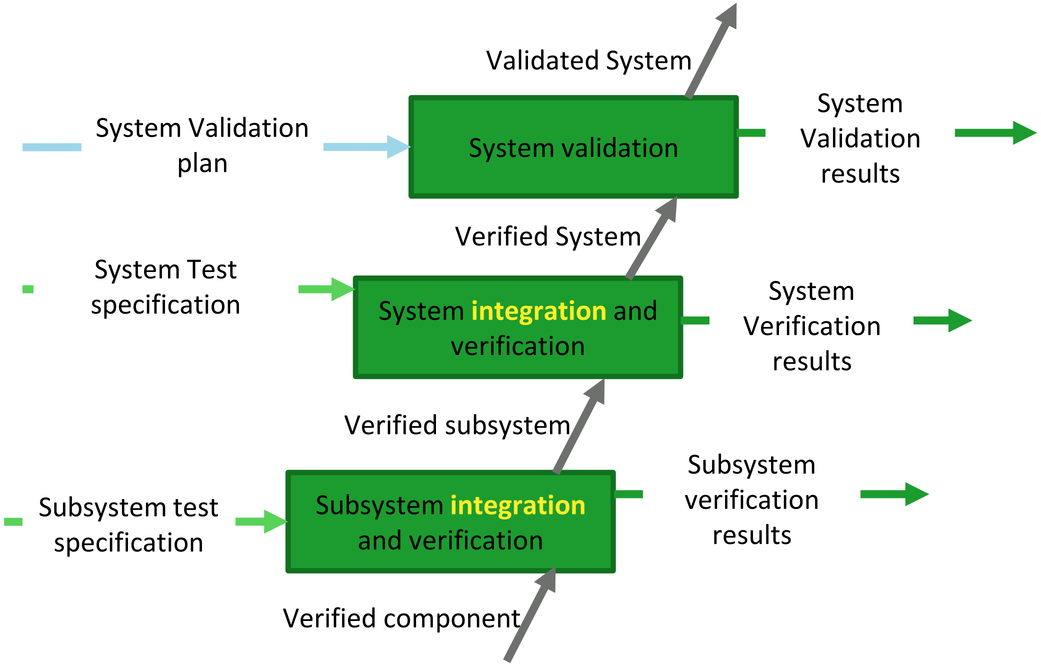
ification of Risk

Control

We only optimized these levels of the product development scope

But what about the other levels?

- **Software is just one of the (verified) components**
- **Integration is where other components a/o hardware, are added**
- **Integration is not an end state, it is a starting point instead**



Validated System
System Validation plan
System validation
System Validation results
Verified System
System Test specification
System integration and verification
System Verification results
Verified subsystem
Subsystem test specification
Subsystem integration and verification
Subsystem verification results
Verified component

Integration

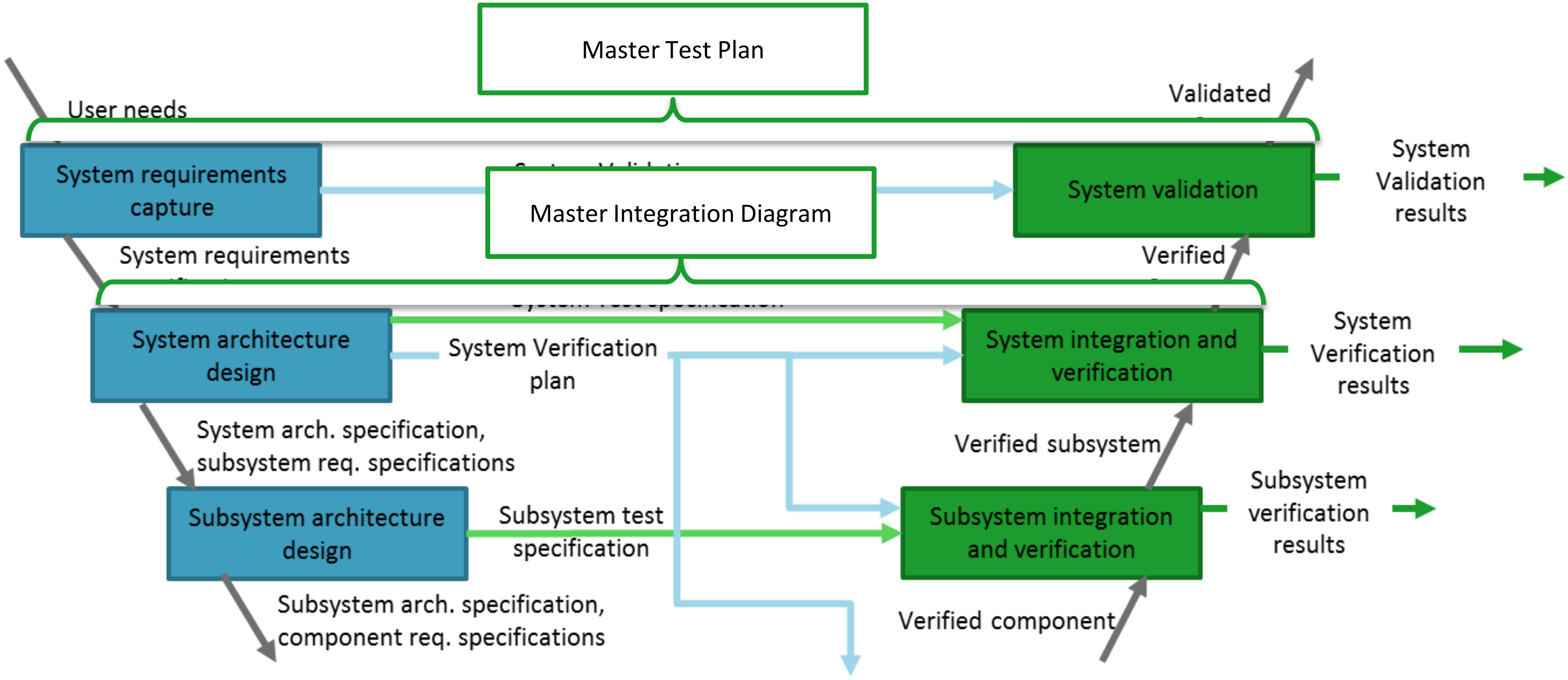
- **An act or instance of incorporating or combining into a whole**
  - *Random House Kernerman Webster's College Dictionary, © 2010 K Dictionaries Ltd. Copyright 2005, 1997, 1991 by Random House, Inc. All rights reserved.*
- **The process of combining components or systems into larger assemblies**
  - *ISTQB Standard Glossary of Terms used in Software Testing - version 2.4*

http://nl.mathworks.com/products/connections/product_detail/service_35832.html

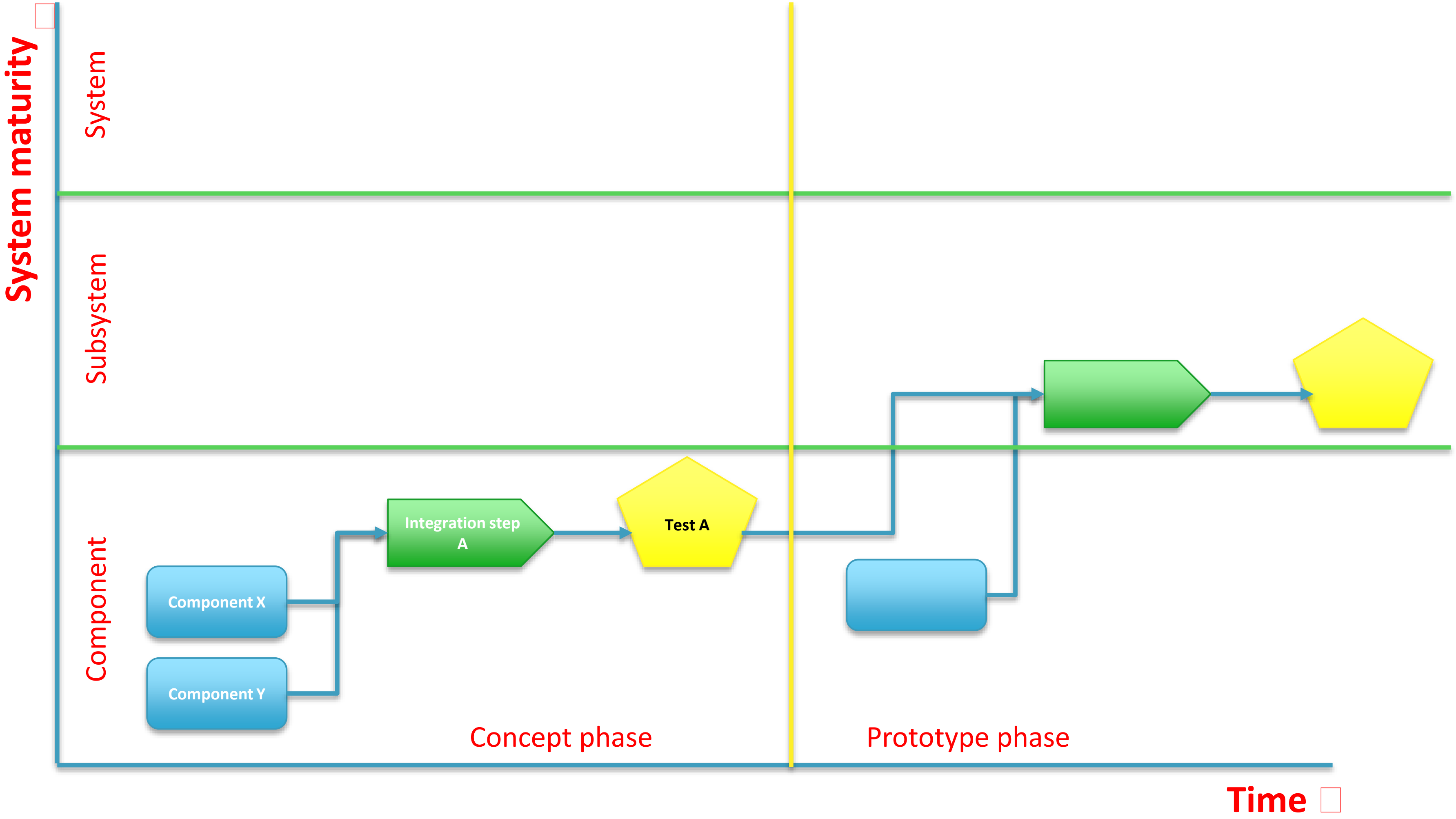
☞ *In testing known as: Hardware-in-the-Loop (HiL)*

Integration

- Asks for a **dedicated** approach
- Is a **creative** activity
- Is not a test activity in itself, but **enables** the verification

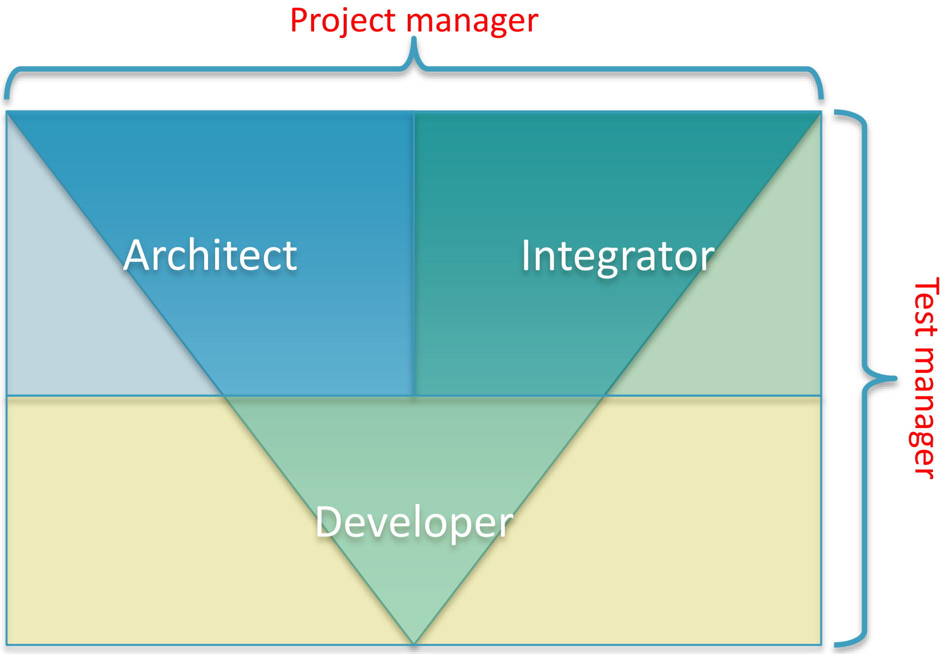

Master Test Plan
User needs
Validated
System requirements capture
Master Integration Diagram
System validation
System Validation results
System requirements
Verified
System architecture design
System Verification plan
System integration and verification
System Verification results
System arch. specification, subsystem req. specifications
Verified subsystem
Subsystem architecture design
Subsystem test specification
Subsystem integration and verification
Subsystem verification results
Subsystem arch. specification, component req. specifications
Verified component

Integration: Dedicated approach


System maturity
System
Subsystem
Component
Component X
Component Y
Integration step A
Test A
Concept phase
Prototype phase
Time

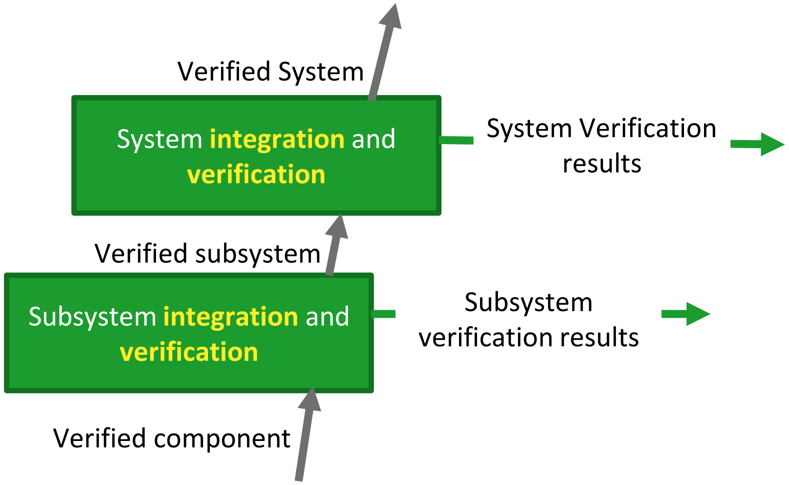
Integration: A creative activity

- **Performed by an integrator**
- **Is an act of building the (sub-)system, focus on make-it-work**
- **Is an act of mitigating risks**
- **Needs accurate planning**
- **Requires mandate to accept or reject component deliveries**


Project manager
Architect
Integrator
Developer
Test manager

Integration: Enables Verification

- **Integration:**
  - Checks if the integrated (sub)system operates
  - Checks if the promised features are delivered
  - Determines to what extend the (sub)system can be operated
  - Determines if and how the risks have been mitigated
  - Determines the required environment for the (sub)system to operate
- **The integrator passes all above information to the stakeholders / test group for efficient verification**
- **Integration is inseparable from the verification to be performed**
- **Verification itself is performed by testers**



Verified System

System **integration** and **verification**

System Verification results

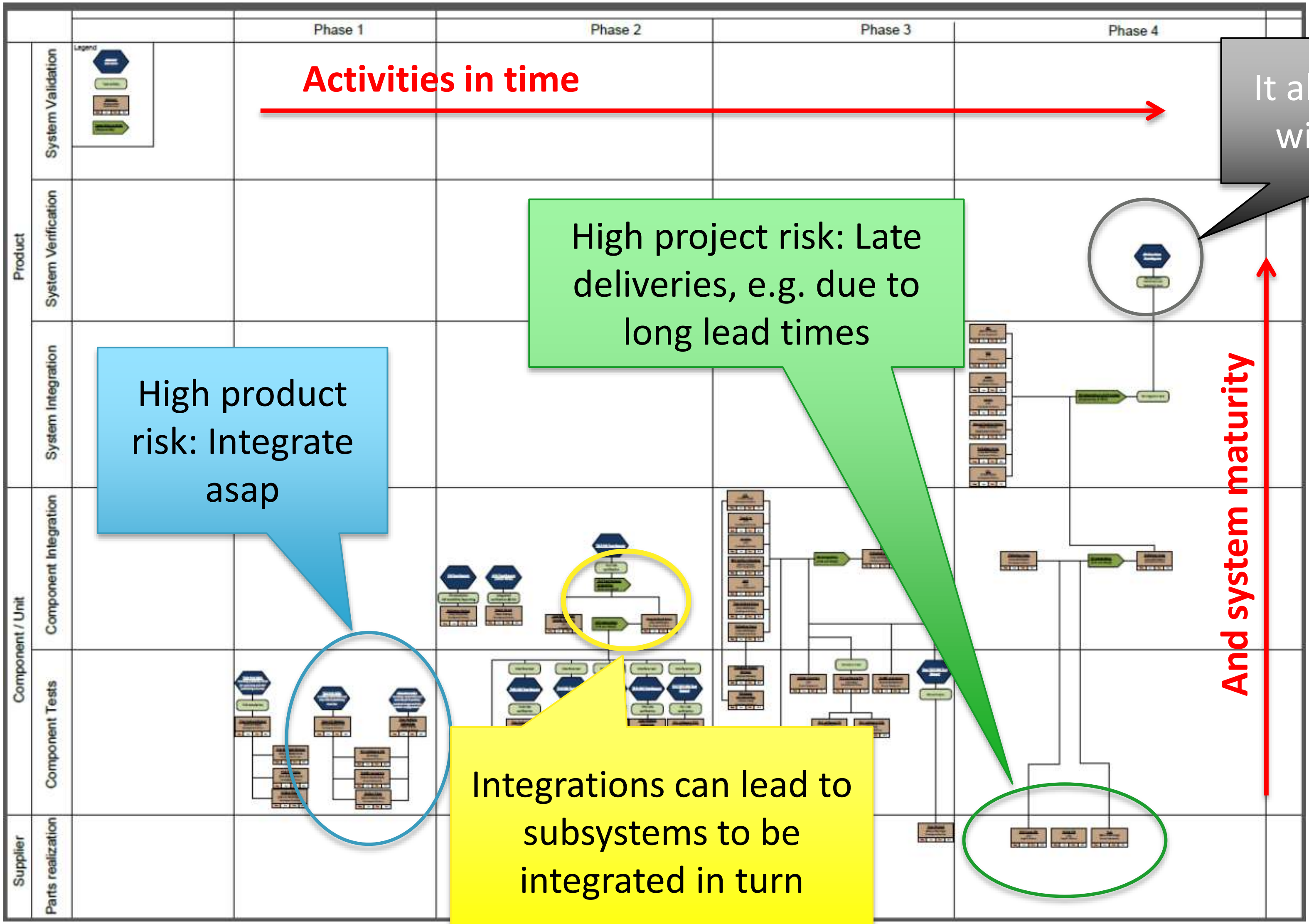
Verified subsystem

Subsystem **integration** and **verification**

Subsystem verification results

Verified component

Master Integration Diagram


Phase 1
Phase 2
Phase 3
Phase 4
Legend
Product
System Validation
System Verification
System Integration
Component / Unit
Component Integration
Component Tests
Supplier
Parts realization

Activities in time

It always ends with a test!

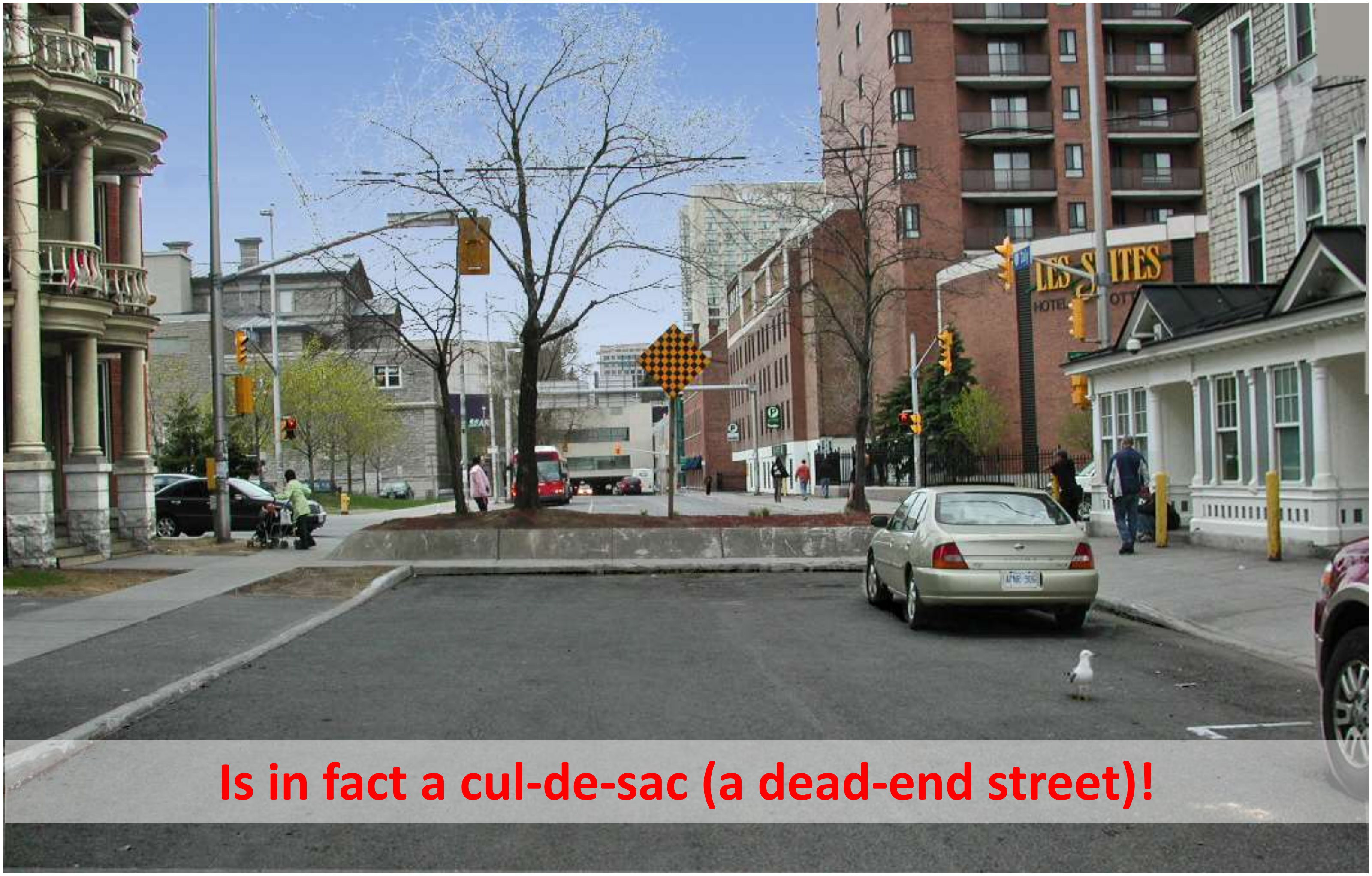
High project risk: Late deliveries, e.g. due to long lead times

High product risk: Integrate asap

And system maturity

Integrations can lead to subsystems to be integrated in turn

Integration without Testing


LES SUITES
HOTEL

Is in fact a cul-de-sac (a dead-end street)!

Integration versus Verification

- Integrators build the system
- Integrators do use some test techniques
  - BVA, EG, ET
- Integration is finished when the (sub)system can be operated
- Integrators need testers to finish the good work

- Testers (try to) break the system
- Testers use many, if not all, test techniques
  - DTT, CFT, CTM, EA, SUT, STT, etc.
- Testing is finished when e.g. the agreed upon coverage is reached
- Testers need integrators to bring them something to test

Integration

- **Forget about Integration Testing**
- **It is either Integration or Testing**

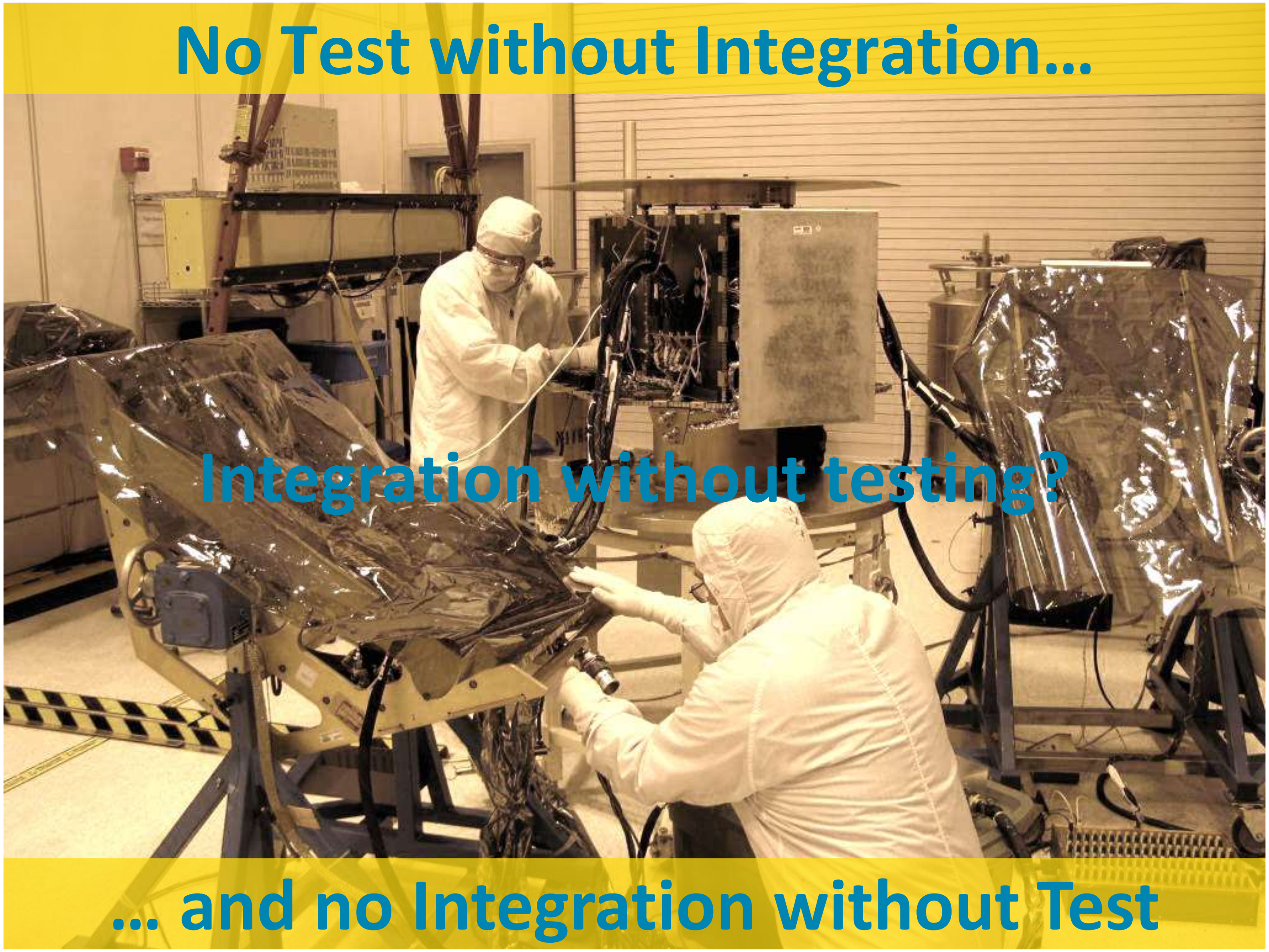
- **A new definition for Integration**
  - *A process consisting of all activities, both static and dynamic, concerned with planning, preparation and evaluation of components and subsystems to determine that they contain specified features and to demonstrate that they are fit for test*
    - *Based upon definition of testing: ISTQB Standard Glossary of Terms used in Software Testing - version 2.4*

Can we optimize any further?

- *Yes!*

- *If you give **both Integration and Test** the position it deserves in your project*

No Test without Integration…



Integration without testing?

… and no Integration without Test

http://www.nasa.gov/mission_pages/ibex/IBEXPhoto2.html

No Test without Integration

André Verschelling
Sr. Test Consultant
Andre.Verschelling@alten.nl
+31 6 526 59743
nl.linkedin.com/in/andreverschelling
@ajfverschelling