PHILIPS

www.philips.com

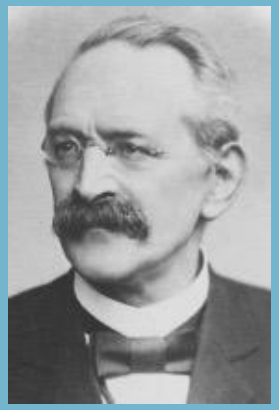
Model Driven Engineering Philips IGT-Systems

Daan van der Munnik
IGT Systems
July 15, 2019

innovation + you

For over 125 years, we have been improving people's lives with a steady flow of ground-breaking innovations

The founding fathers of Philips



Frederik

Gerard

Anton

We strive to make the world
healthier and more sustainable
through innovation

We're aiming to **improve the lives** of

3 billion people a year

by 2030

We are on a journey to transform Philips into a health technology leader

Technology	Health technology
Product	Solution
Transaction	Relationship

Addressing the Quadruple Aim

Improved patient experience

Improving the patient experience of care (including quality and satisfaction)

Better health outcomes

Improving the health of individuals and populations

Improved staff experience

Improving the work life of health professionals

Lower cost of care

Reducing the per capita cost of healthcare

Winning propositions

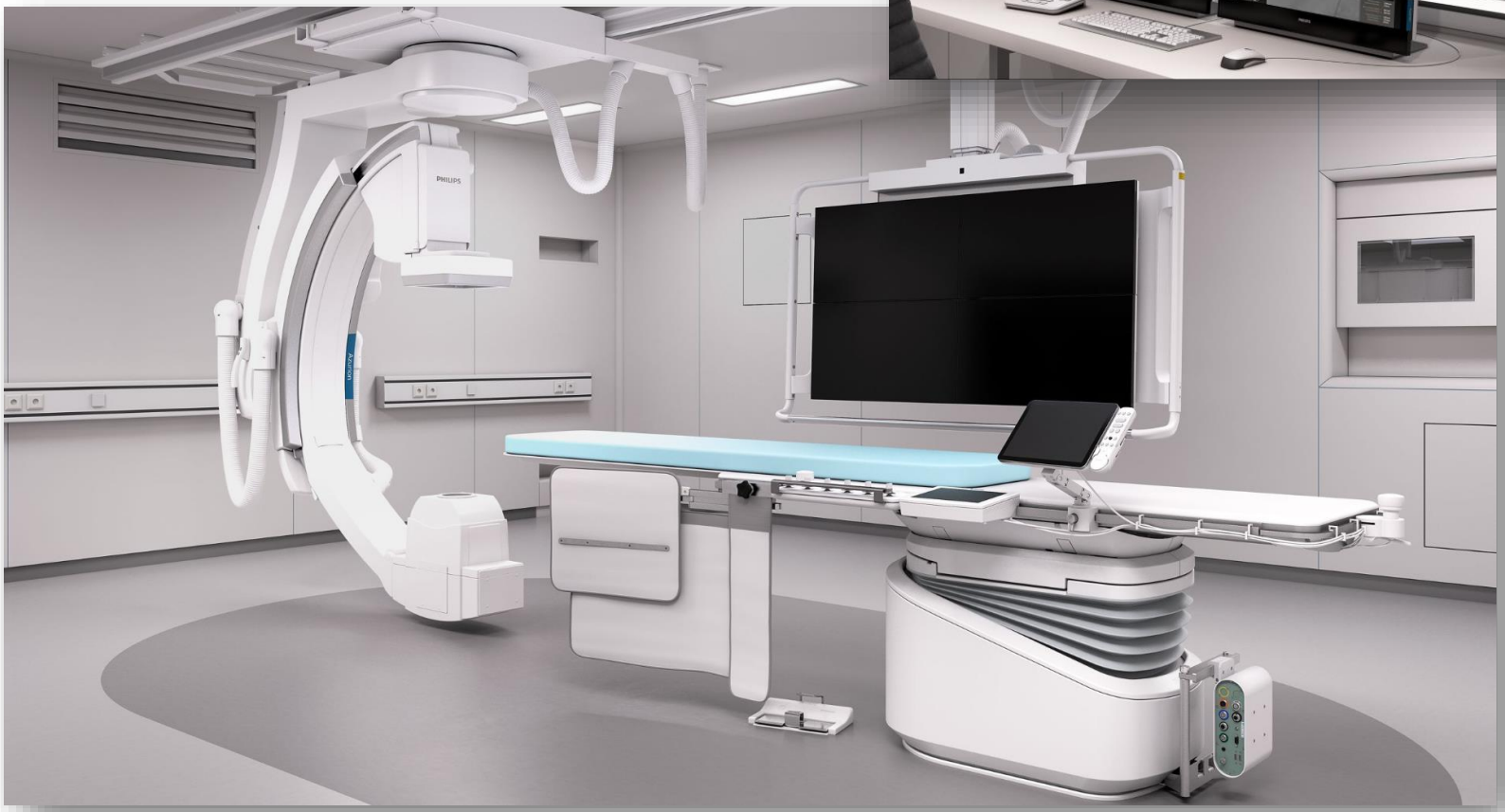
>60% of sales from leadership positions[1]

Diagnosis & Treatment	**Ultrasound** Global leader	**Image-guided therapy systems** Global leader	**Image-guided therapy devices** Global leader	**Diagnostic imaging** Global top 3	**High-end radiology and cardiology informatics** #1 in North America
Connected Care	**Patient monitoring** Global leader	**ICU telemedicine** #1 in North America	**Personal emergency response** #1 in North America	**Respiratory care[2]** Global leader	**Sleep care** Global leader
Personal Health	**Male grooming** Global leader	**Oral healthcare** Global leader	**Mother and child care** Global leader	**Domestic appliances** Air, #1 in China	

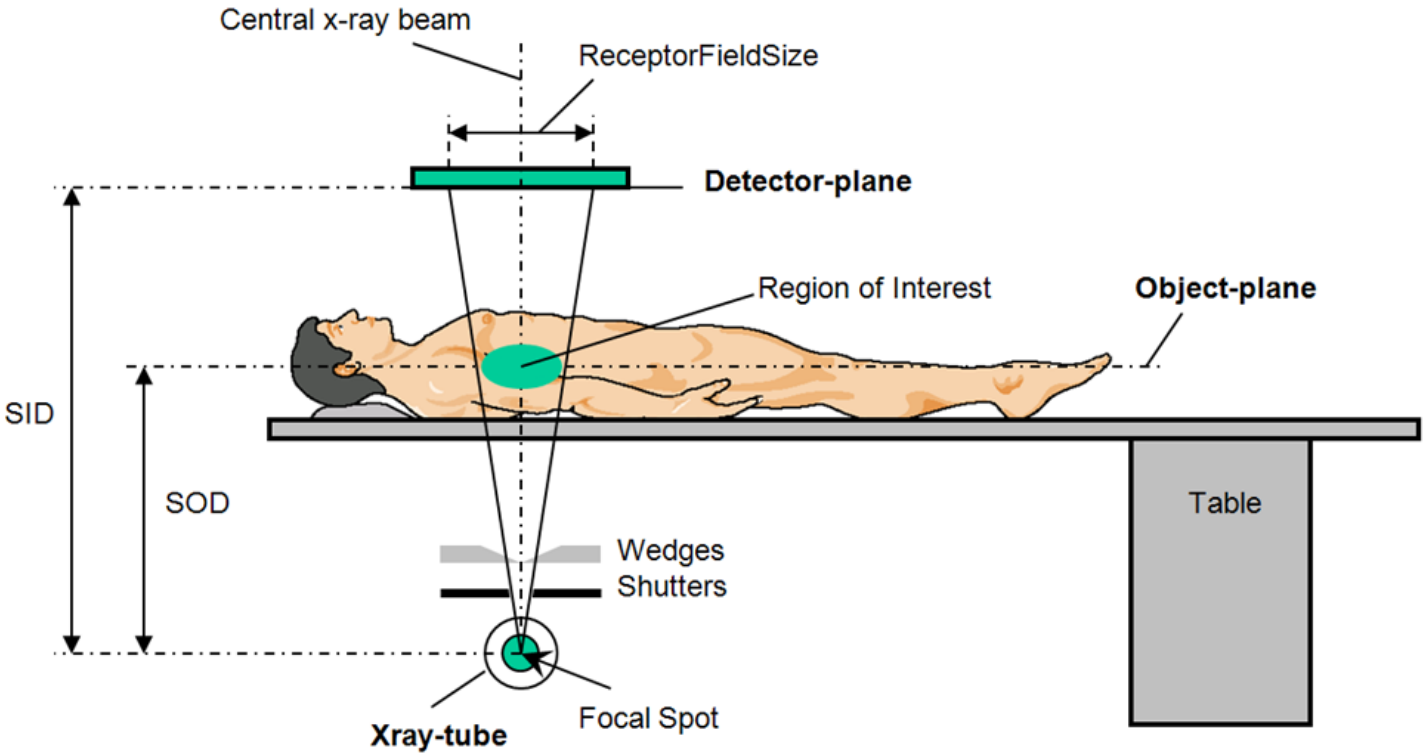
1 Leadership position refers to #1 or #2 position in Philips addressable market. Effective Q1 2019, Philips' reporting structure changed as per announcement of January 10, 2019.
2 Based on non-invasive ventilators for the hospital setting

Source: GfK, Nielsen, Euromonitor, Frost and Sullivan, Home Healthcare TBS, PCMS market insight

Philips Azurion



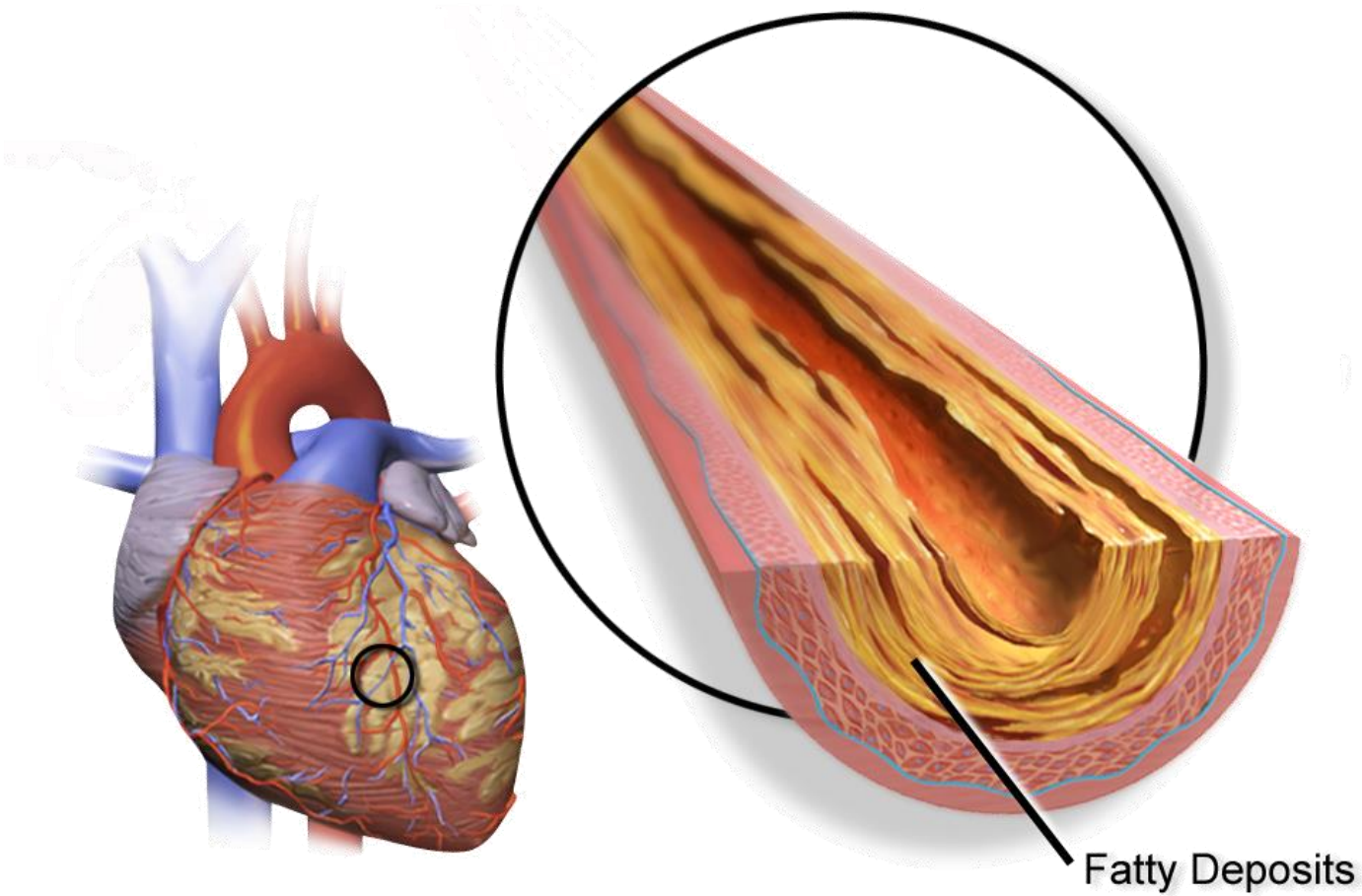
PHILIPS


Central x-ray beam
ReceptorFieldSize
Detector-plane
Region of Interest
Object-plane
SID
SOD
Table
Wedges
Shutters
Xray-tube
Focal Spot

PHILIPS

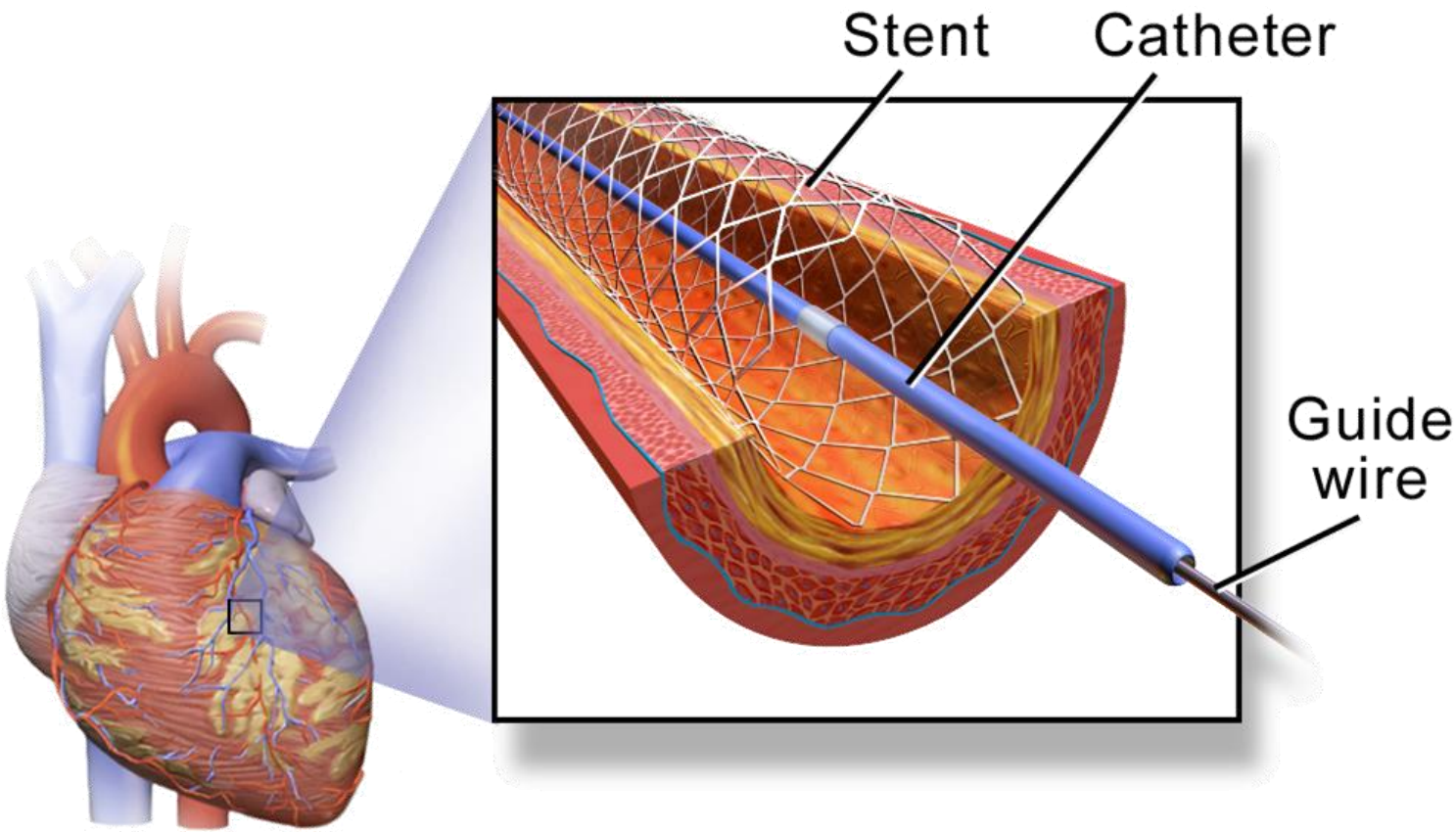
PHILIPS

Partially Blocked Coronary Artery



PHILIPS

Stent in Coronary Artery



PHILIPS

Applications Presets Examination Room System Help PHILIPS

Acquisition

Live | PATIENT, Name | ID 13579ZA24 | DOB 07-Mar-1957

X-ray Settings
Series
Processing

X-ray Protocol
Coronary (ProcedureCard)

Patient Orientation

Fluoroscopy
Roadmap
On Off
Smart Mask
Mode
Navigate
Reset Device
Exposure
Injector

Series 1
Image 41
01-Dec-2017, 7:36 AM

LAO 22°
CAUD 22°

AO 120 / 82 (97) 80

AIR REST
ECG> 12 LEAD
AO 120 / 82 (97)

118 / 78 92 48 36.8 °C 98 %

VOLCANO IVUS 12345678 Case Explorer Frames: 10 Loops: 2

VL1: LAD Pre-Stent (PAUSED)

Area: 19.5mm²
Min Dia: 4.9mm
Max Dia: 5.1mm

Differences
14.5mm² (74.2%)

Area: 6.0mm²
Min Dia: 2.3mm
Max Dia: 2.9mm

Live Diameter Draw Dots Delete

Patient Home W/L End Case Archive Retrieve

Press Measure for Autoborders or Select from Measurement Toolbar

kV 40 mA 200 ms 100 | 0° 0° | SID 39 inch FD 22.0 inch | Isocenter Position | X-Ray frontend is stubbed | Left Coronary 5 fps Medium | Time Left 0 s | K 0 mGy | Total K 0 mGy | DAP 0.0000 Gy·cm² | Total Fluoro 0.0 min | 1:27 PM

Challenges

In system of systems development

PHILIPS

#1: Patient (and operator) safety

- X-ray / dose
- Moving geometry
- Live Imaging functionality
  - interventional procedures
- Hand-eye coordination
  - latency

End-to-end CTQ's
(critical-to-quality)

PHILIPS

Medical Device Regulations

- External
  - FDA 21 CFR 820 [1] (Quality System Regulation)
    - Submission
  - IEC 62304 (Medical device software – Software life cycle processes)
- Internal
  - Business Management System
- Audits
  - Say what you do, do what you say
  - Prove it, improve it

Evidence required

(1) https://www.accessdata.fda.gov/scripts/cdrh/cfdocs/cfcfr/CFRSearch.cfm?CFRPart=820&showFR=1

System of systems

- Many interconnected subsystems
  - Developed by Philips
  - Contracted out
  - Off-the-shelf
- Specialist teams, **different domains & disciplines**

Complexity, Integration issues

- Subsystems brought together during **integration and test phase**
- Discovering **interface problems** at this stage is extremely **costly**
- Often caused by:
  - Ambiguous specifications
  - Undocumented usage
  - Unintended changes to functionality and/or system-level timing

- Relationship between subsystems should be made clear early in design phase
- Requires an **unambiguous definition of the interfaces**
  - i.e. protocols, timing, data ranges and functionality

PHILIPS

Long lifetime

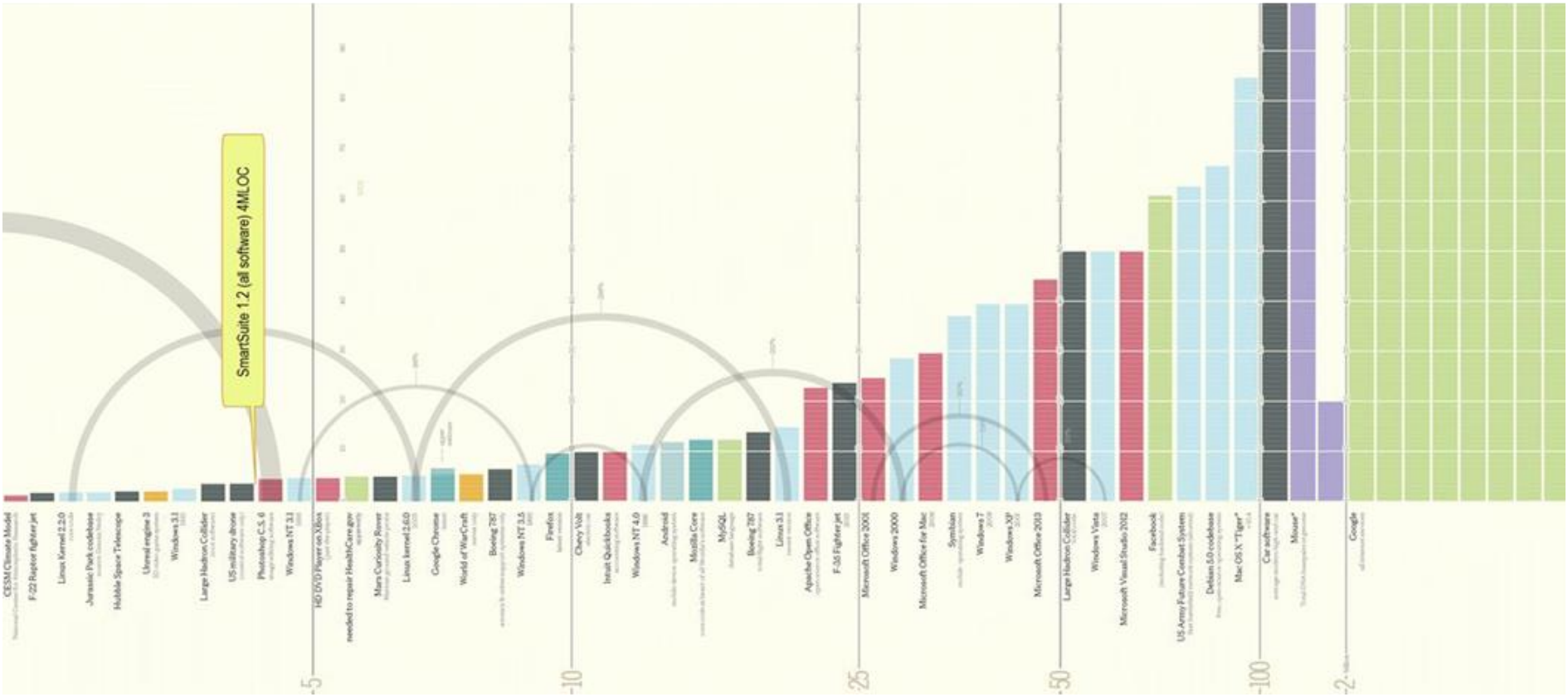
- Support obligation
  - For each system we sell, we implicitly commit to sustain it for **>12 years**

- Hardware lifecycle
  - Fact:
    - Each component shipped as part of the system, has a certain probability to break during the life time of the system
  - Ideal situation:
    - Form fit function replacements /upgrades
      - Replace a FRU without full install or interface failure

Need for: form-fit-function replacements/upgrades

PHILIPS

Code size

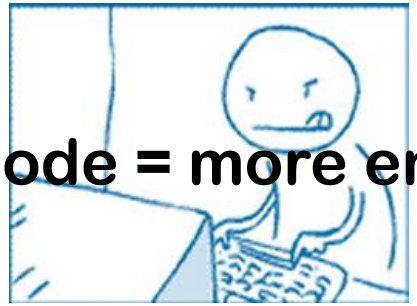


PHILIPS

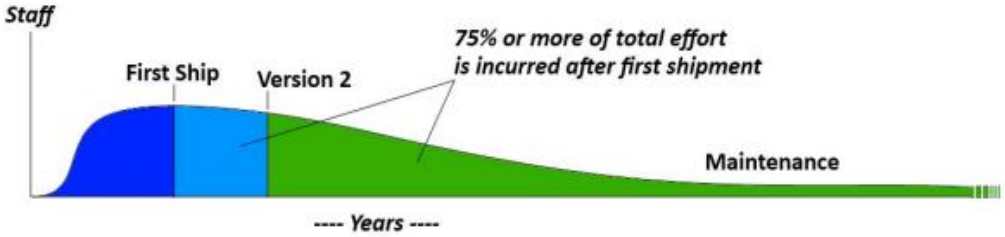
Basic laws of software maintenance

errors =(more code)^2

more code = more engineers



- Each full-time developer can maintain 50KLoc <u>or</u> create 10KLoc per year [1]
- 75% of the total effort is incurred after the first shipment [1]



[1] Wayne Lobb e.a "Software Development and Maintenance Effort/Cost Models", Foilage whitepaper

PHILIPS

Legacy code

(code older than 2 weeks ☺)

- Software deteriorates over time
  - Over-production
    - It takes skills, time and review to create concise software.
  - Over-abstraction
  - Over-processing
  - Unnecessary glue/interface layers

- The maintenance trap
  - More engineers -> more code -> more maintenance -> more engineers
    - Over time teams will slow down to a halt
  - Maintenance = 75+% of SW development cost

PHILIPS

Challenges

summary

- Patient safety & Medical device regulation
  - End to end CTQs
  - Evidence required

- System of systems
  - Complexity, integration issues

- Long lifetime
  - Need for form-fit-function replacements/upgrades

- (Legacy) code size
  - More code = more engineers = more $

PHILIPS

Approach

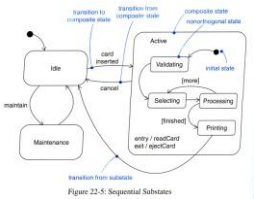
- **Strong focus on SW quality**

First time right, (prevents expensive releases)

- **Managing Code-down**

Reducing (legacy) code size

- **Model Driven Engineering**

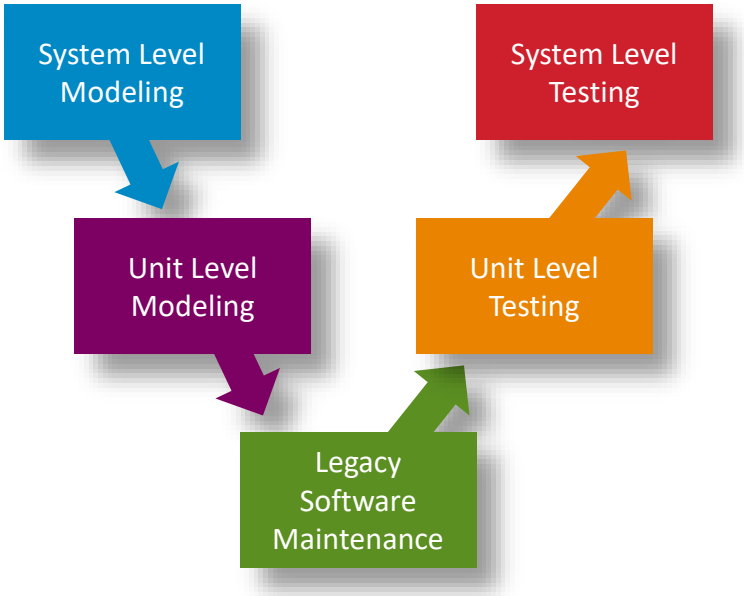


Addresses <u>both</u> first time right & code size as well as complexity & integration issues

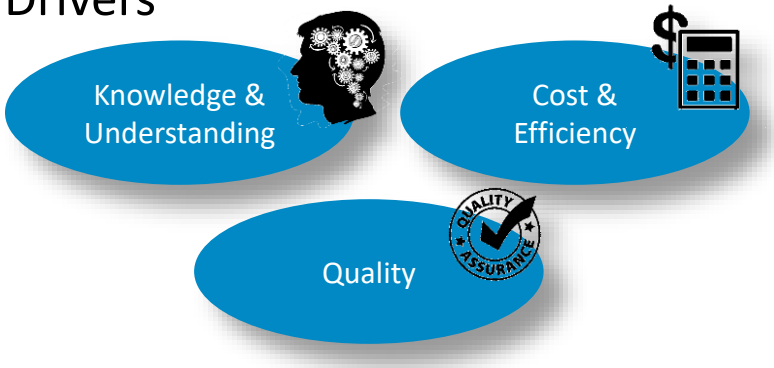
PHILIPS

Modeling Roadmap

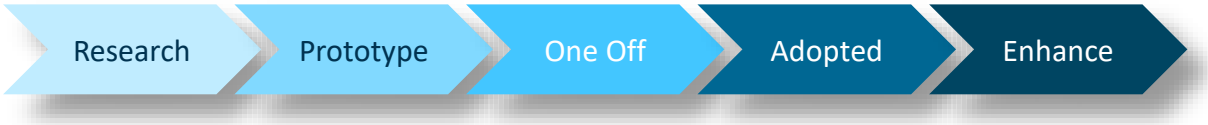
Categories



Drivers



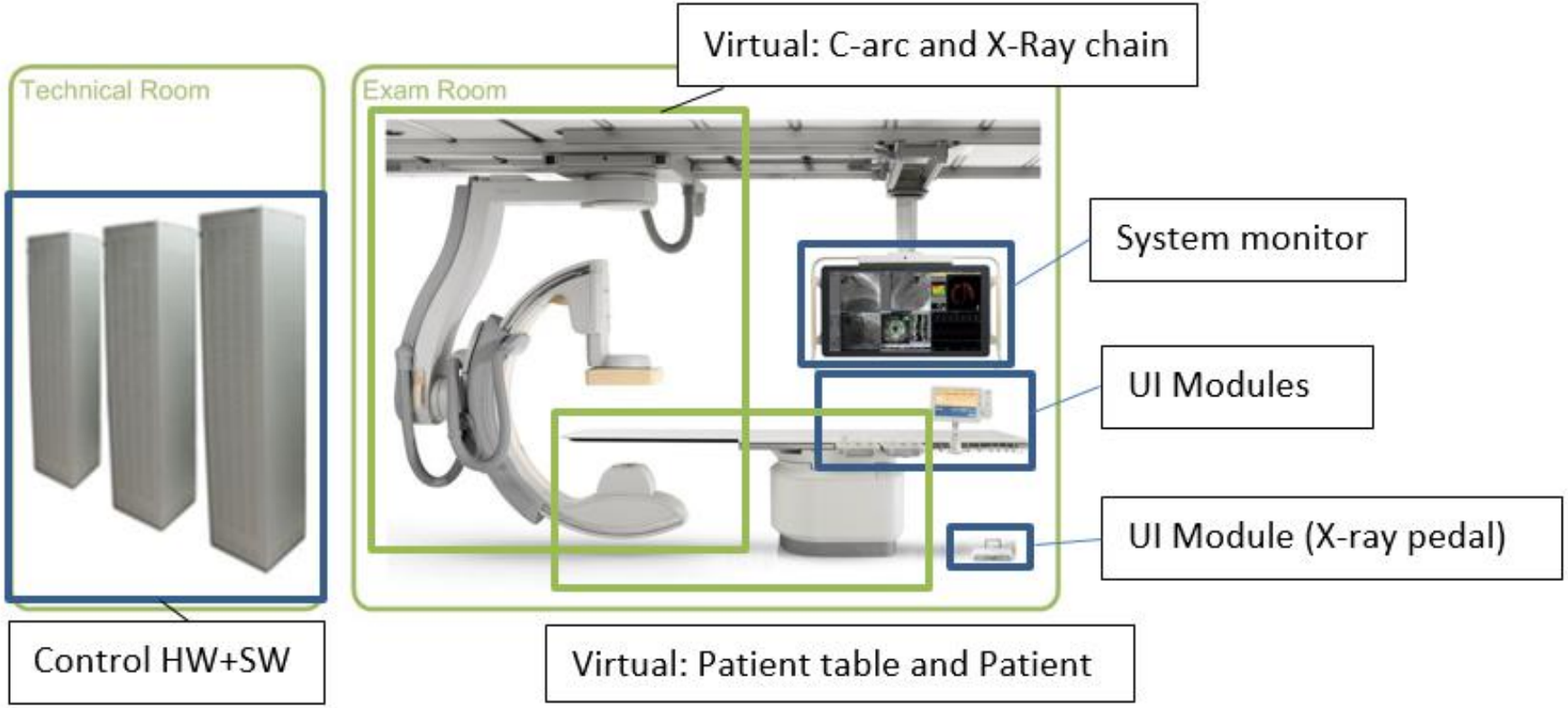
Maturity



PHILIPS

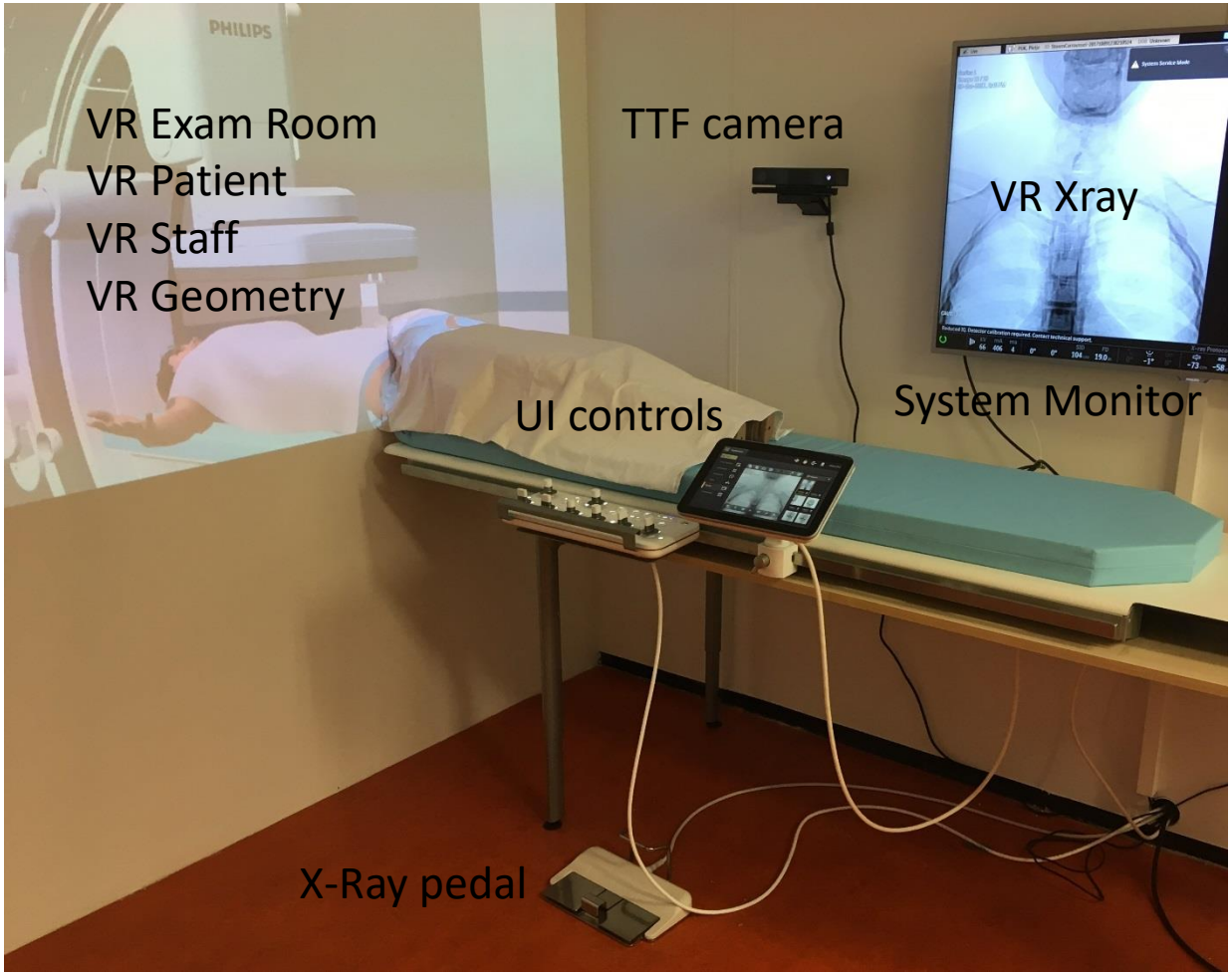
System Level Modeling

Virtual Cathlab

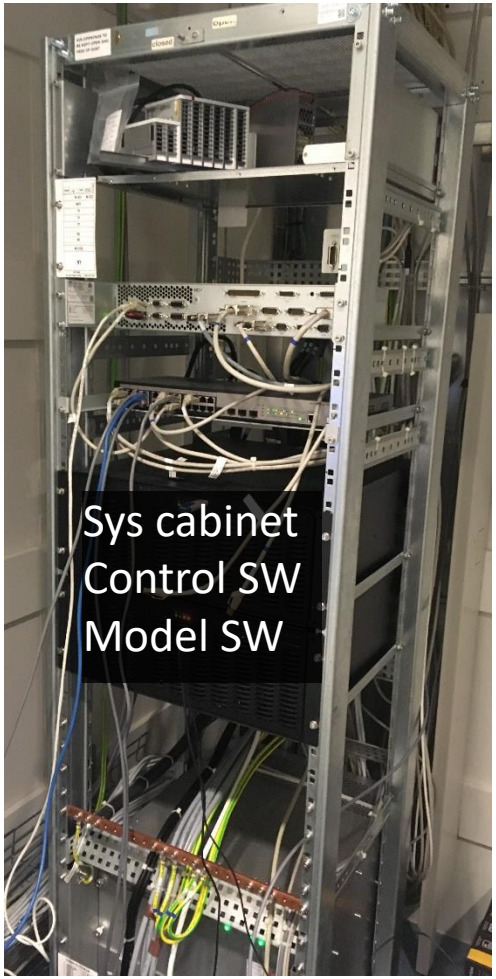


PHILIPS

Exam Room



Technical Room



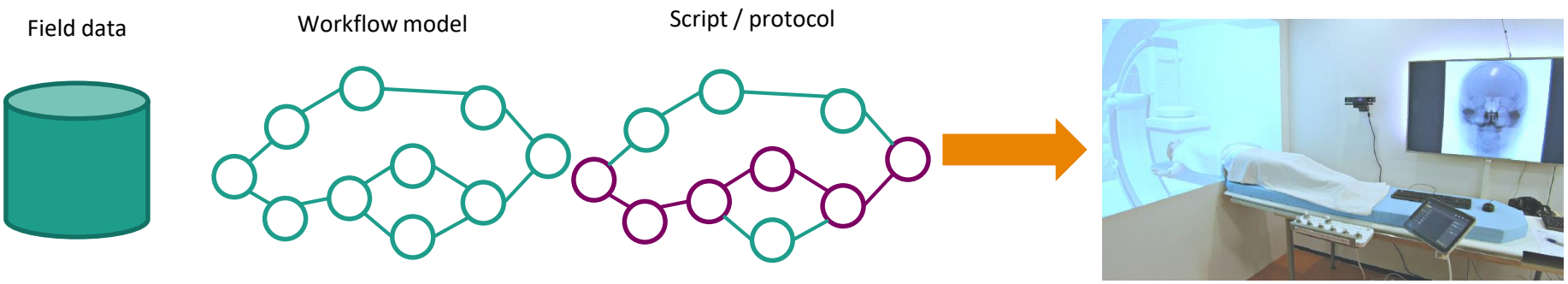
PHILIPS

System Level Modeling

Clinical workflow modelling

and Usability Tests

- Use of partly simulated environment for Usability Testing
- Clinical Workflow modelling to create field based usability test protocols
  - Based on:
    - Machine logging
    - Procedure logging
    - Location data



PHILIPS

Unit Level Modeling

Unit Level Modeling

- Interface modeling and conformance checking
  - Define, monitor and test interfaces across subsystems
  - “learn” the characteristics of existing interfaces
  - Model-based testing generated from interface models

- Extend the use of MDD using Dezyne
  - Model all new complex state behavior using Dezyne
  - Replace existing manual coded state behavior with Dezyne (where possible)

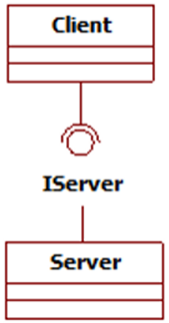
PHILIPS

Unit Level Modeling

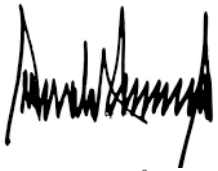
Interface management

ComMA

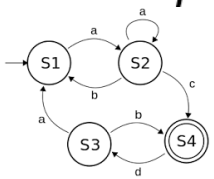
- **Rigorous interface management**



- Signature



- Behavior



- Timing constraints

- Data constraints

- Semantics

- Throughout the lifecycle
  - Design
  - Implementation
  - Test
  - LCM

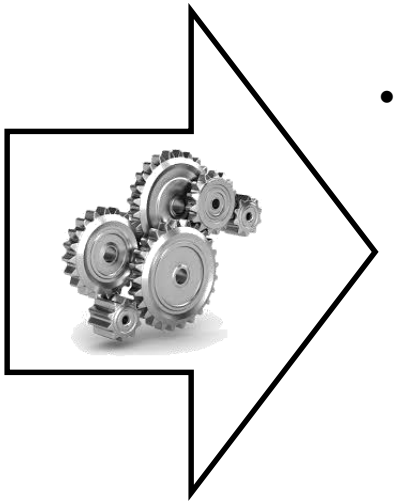
PHILIPS

Unit Level Modeling

ComMA

Tool suite

- In:
  - Signature
  - Behavior
  - Constraints (timing & data)



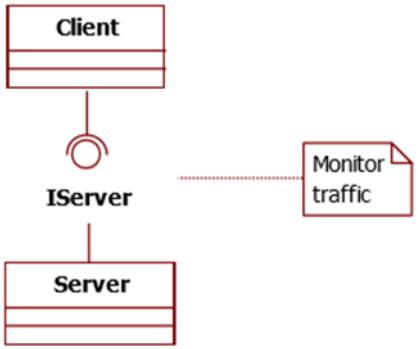
- Out:
  - Visualization
  - Documentation
  - Interface proxy code
  - Executable model for simulation
  - Runtime monitoring
  - Model based testing suite
  - Client/server simulator/stub

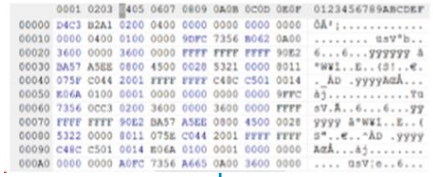
- Co-owned & created by Philips & TNO-ESI
- **Free** to use and download (for non competitors)
  - We are interested in strengthening the ecosystem

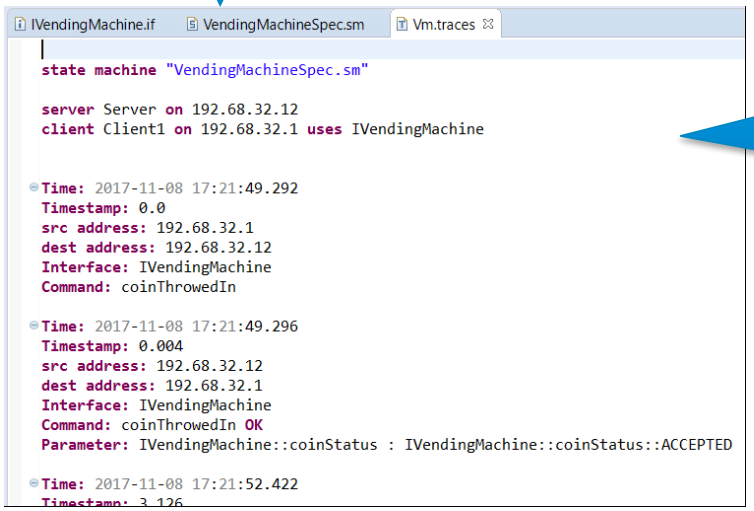
PHILIPS

Unit Level Modeling

Runtime monitoring







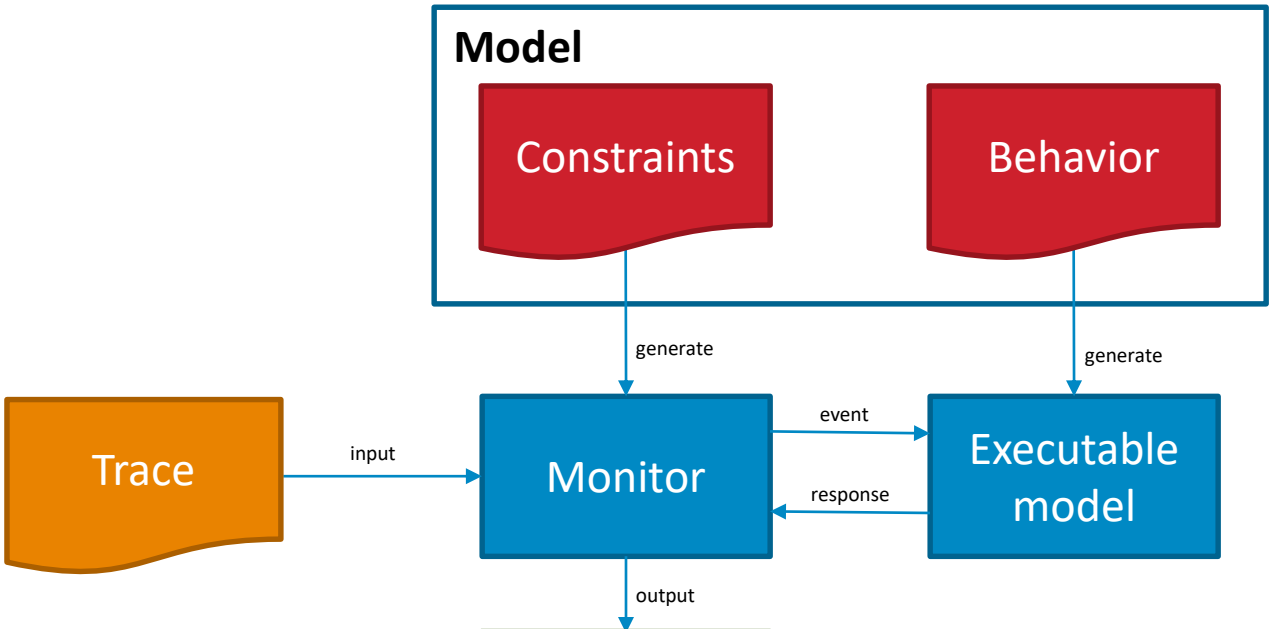
Technology independent trace language

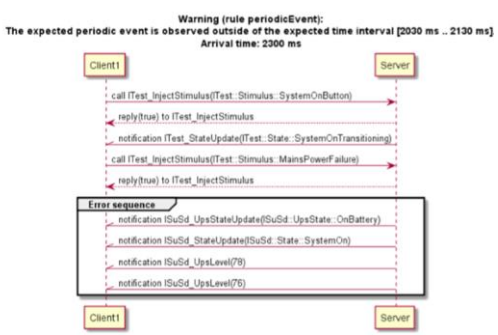
PHILIPS

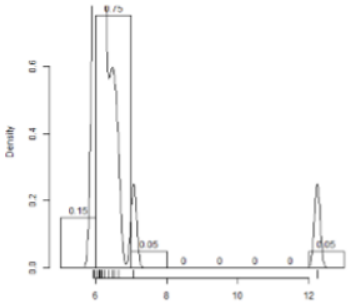
Unit Level Modeling

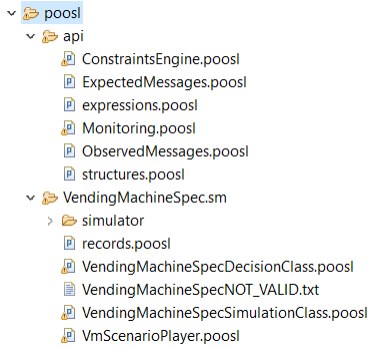
Executable model for simulation

Interface conformance checking









PHILIPS

Unit Level Modeling

Work in progress / Roadmap

- **Test case generation** from interface definitions
- Supporting definition of new interfaces and development
  - Define **Scenarios**
  - **Simulation** of interface
  - Smart **Stubs**
- Modeling existing interfaces - **reverse engineering**
- Generate **formal models** – enable formal analysis
- From interfaces to **components & (sub)systems**

PHILIPS

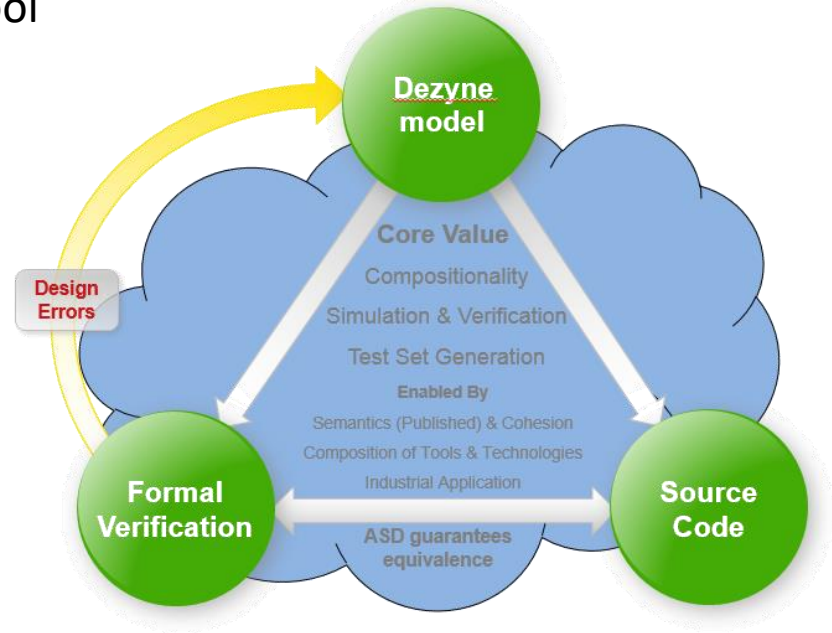
Unit Level Modeling

Dezyne

(Verum)

- Model-driven software engineering tool
  - Modeling language
  - Simulation
  - Formal verification
  - Code generation
- For state based, event driven or concurrent software systems

- Used in:
  - ACQ subsystem (ASC/BEA)
  - iEngine (roadmapping, service)



PHILIPS

Legacy Software Maintenance

- Software **rejuvenation** using compiler techniques
  - Model extraction from legacy components to improve human understanding
  - *Static* dependency visualization & refactoring tooling

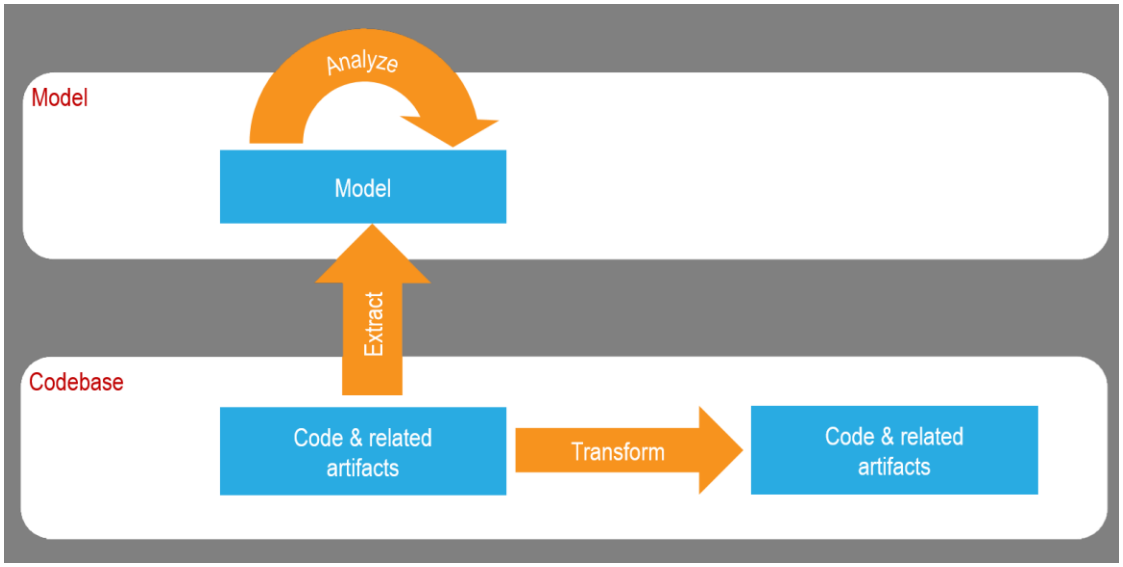
- Re-engineering *dynamics* of legacy software towards MDSE
  - By analyzing source code (model extraction)
  - By analyzing runtime traces (model learning)

PHILIPS

Rejuvenation

Static dependency visualization & refactoring

- Develop scalable techniques for improving software quality (architecture, code, etc.)
- Three step approach:
  - Extract
  - Analyze
  - Transform



PHILIPS

Extract

- The extraction tooling extracts facts from:
  - Build infrastructure related files, e.g., Visual Studio solutions/project files
  - Source files, e.g., C/C++/C# files and COM-related (.idl) files
  - File systems, e.g., the disk layout of the source tree

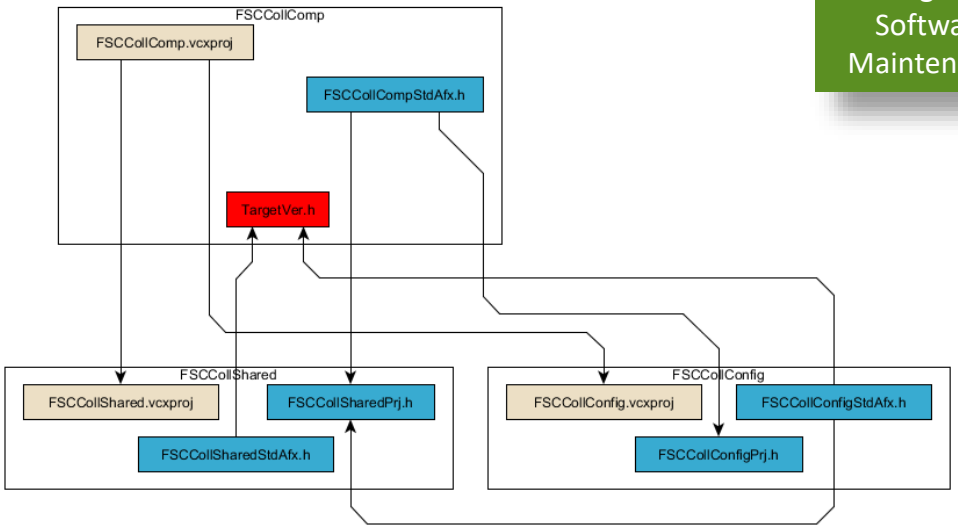
- The extracted model is stored as a graph database

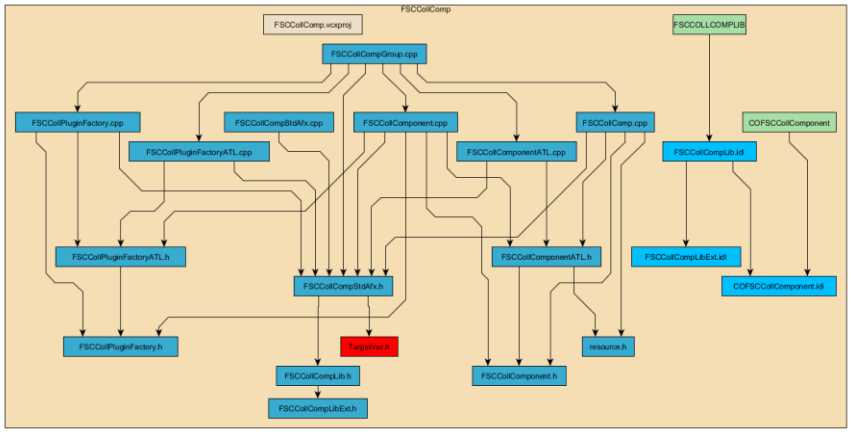
Legacy Software Maintenance

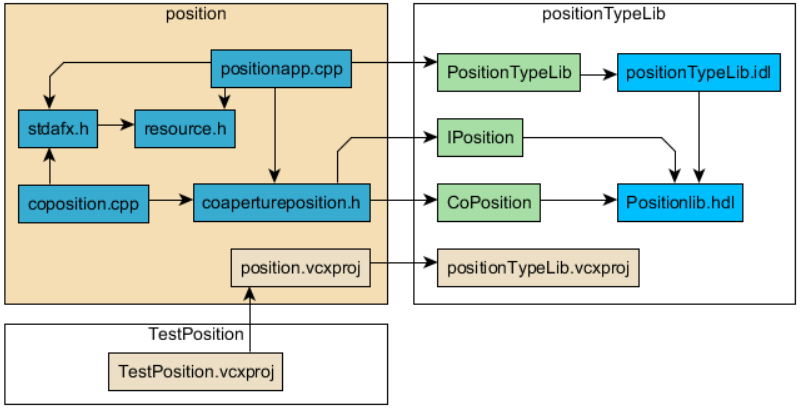
Analyze

yEd

- Customizable visualizations of dependencies





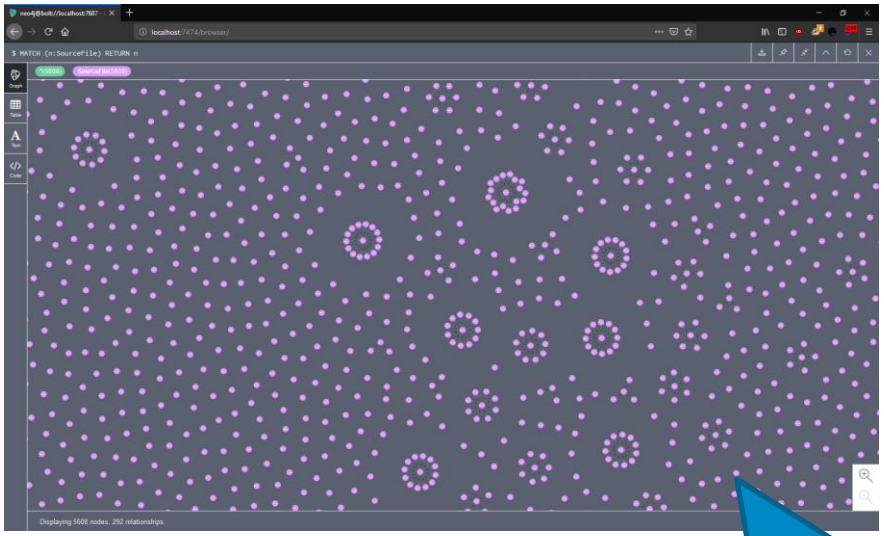


PHILIPS

Analyze

Neo4J

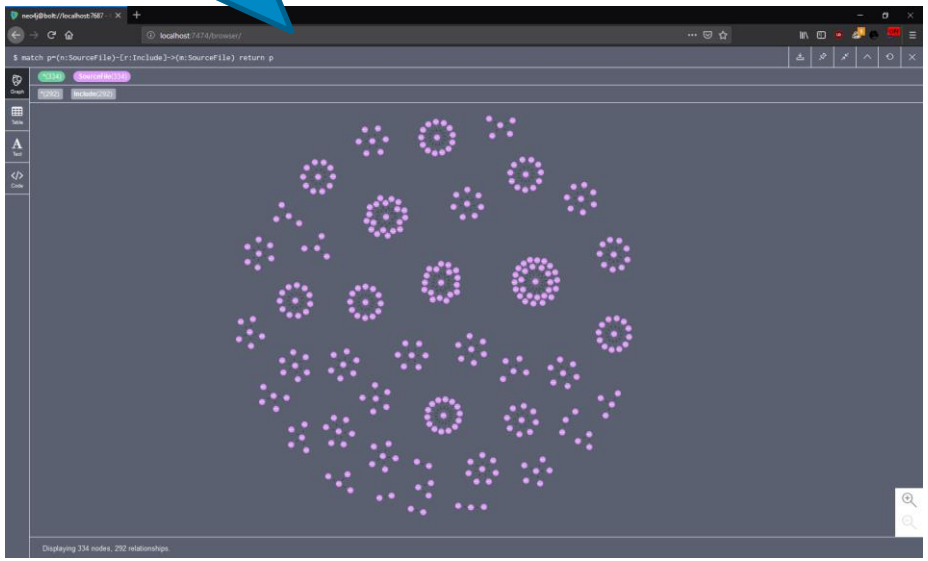
- Ad-hoc queries



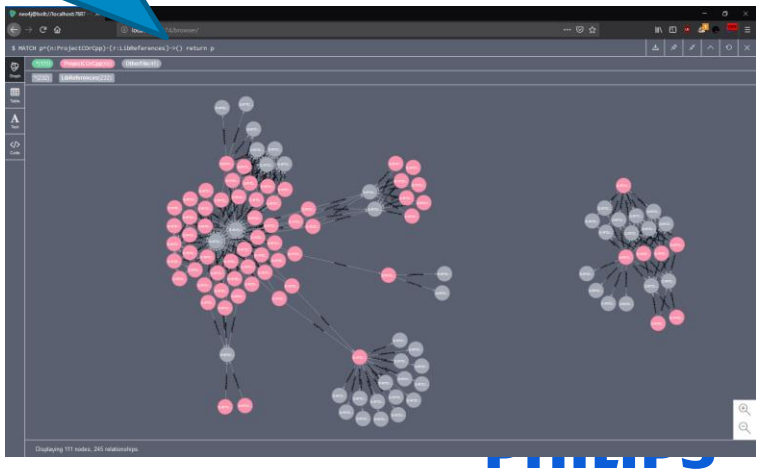
Legacy Software Maintenance

All sourcefiles

Sourcefiles including sourcefiles



Library references



Legacy Software Maintenance

Analyze

Design Structure Matrix

$root			1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Utility	U3	1	.	2				2						13				11	6
	U2	2		.		13	13		3	4						2			9
	U1	3			.	10	10	2		2	2	9	4	13		3			2
Presentation	P5	4				.													
	P4	5		1			.												
	P3	6		1				.									1		
	P2	7		1					.										
	P1	8		1						.									
Data	D3	9				4					.								
	D2	10		1								.							
	D1	11		1									.						4
Application	A6	12												.					
	A5	13													.	1			
	A4	14													5	.		2	
	A3	15		1													.		1
	A2	16		1			1		1									.	
	A1	17		1						1									.

Same rows and columns

Identity line

Dependency

Hierarchy

PHILIPS

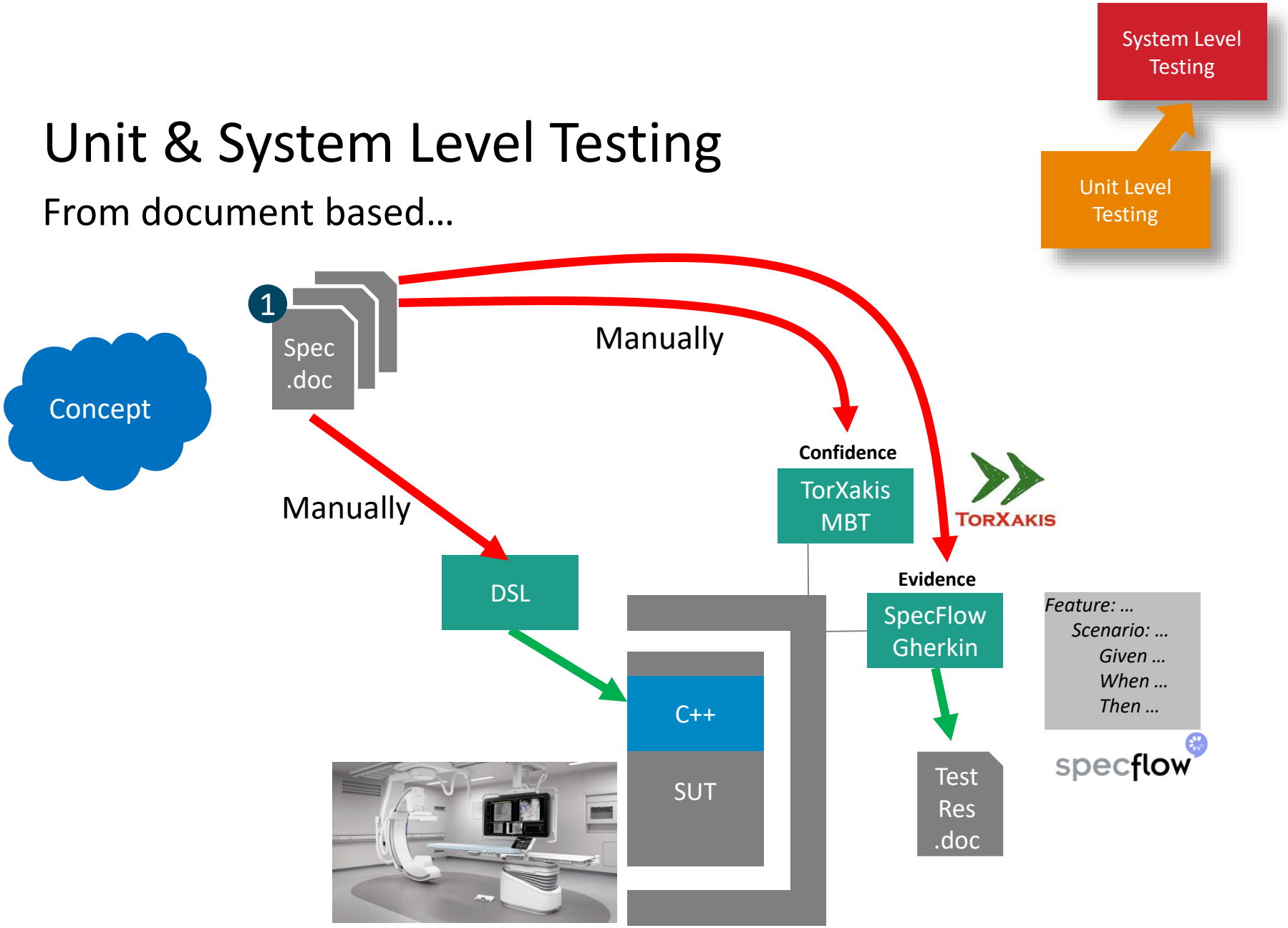
Transform

- Use-case
  - Remove .cpp “group” files including other .cpp files (faster builds, easier navigation, code-down)
- Results
  - Significant improvement in productivity (4 hours vs. 1 week)
  - Amount of repetitive work is reduced
  - Part that requires creative thinking remains

- Also using queries for architecture guarding
  - Integrated in CI/CD pipeline

Unit & System Level Testing

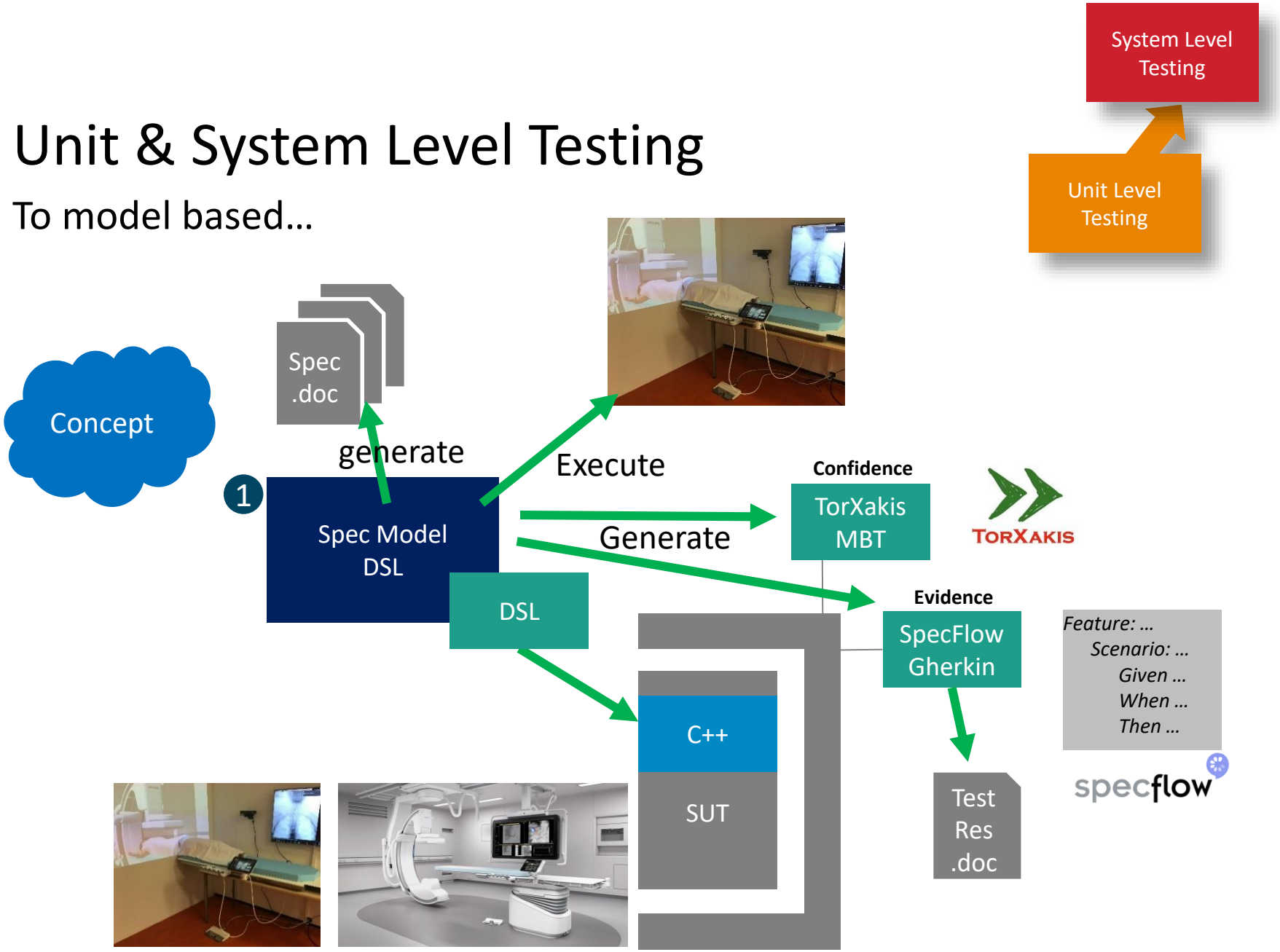
From document based…



PHILIPS

Unit & System Level Testing

To model based…



PHILIPS

Summary

- Model Driven Engineering helps us at all levels to:
  - Increase Quality
  - Manage Complexity & Integration issues
  - Reduce & Manage legacy code

- Addressing the Quadruple Aim:

Improved patient experience

Better health outcomes

Improved staff experience

Lower cost of care

PHILIPS

PHILIPS