SAAB
ONEFACT CONFERENCE



Wilhelm Isoz, Saab Aeronautics

This document and the information contained herein is the property of Saab AB
and must not be used, disclosed or altered without Saab AB prior written consent.

IN 1937 WE TOOK OFF

- A history from the 17th century through Alfred Nobel (Bofors) and the shipyard in Karlskrona (Kockums)
- In 1937, Saab was founded to protect Sweden's borders and its people
- Born smart – as a small country, we were forced to arm ourselves with good and cost-effective equipment
- On our journey we created Sweden's computer, missile and space industries



OUR **PERFORMANCE** IS STRONG

2016

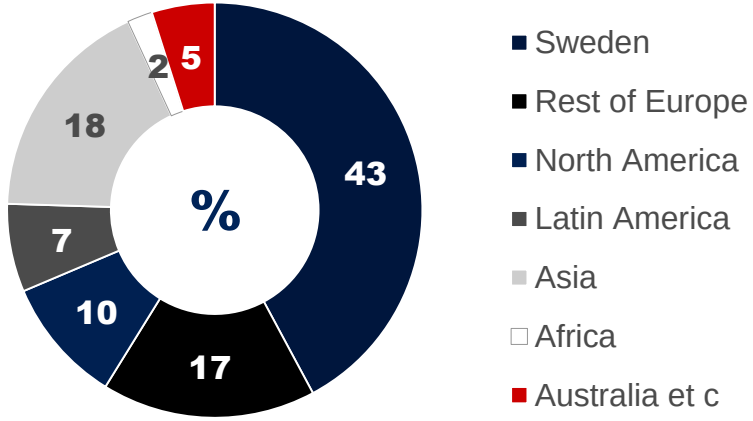


- Historic year with launch of three new platforms:
  - Launched a new generation Airborne Early Warning system, GlobalEye
  - Rolled out new generation of Gripen
  - Rolled out the trainer aircraft for the US Air Force T-X-programme; the T-X aircraft flew in December
- Sales growth in line with long-term financial goal of an organic growth p.a. of 5 per cent

Order backlog, MSEK	**107,606**
Sales, MSEK	**28,631**
Operating income*, MSEK	**1,797**
R&D, share of sales %	**26**

* EBIT

Geographic distribution of sales



HOW WE ARE **ORGANISED**

BOARD

CEO

MARKET AREAS

- Asia Pacific
- Europe
- Latin America
- Middle East & Africa
- North America
- Our headquarters…

GROUP FUNCTIONS



BUSINESS AREAS

- Aeronautics
- Dynamics
- Industrial Products and Services
- Kockums
- Support and Services
- Surveillance

OUR BROAD **OFFERING**



MODELING@SAAB

SAAB MODELING DAY

- First modeling conference within Saab
- Spring 2017
- 97 people attended
- Representatives from all business areas

- Conclusions from the conference
  - MBSE is well spread thought the company
  - Focus on system engineering
  - Codegeneration mainly applied within Dynamics and Aeronatuics

LEARNINGS FROM SAAB MODELING DAY

ALWAYS START WITH THE

WHY

THE WHAT AND HOW COMES LATER

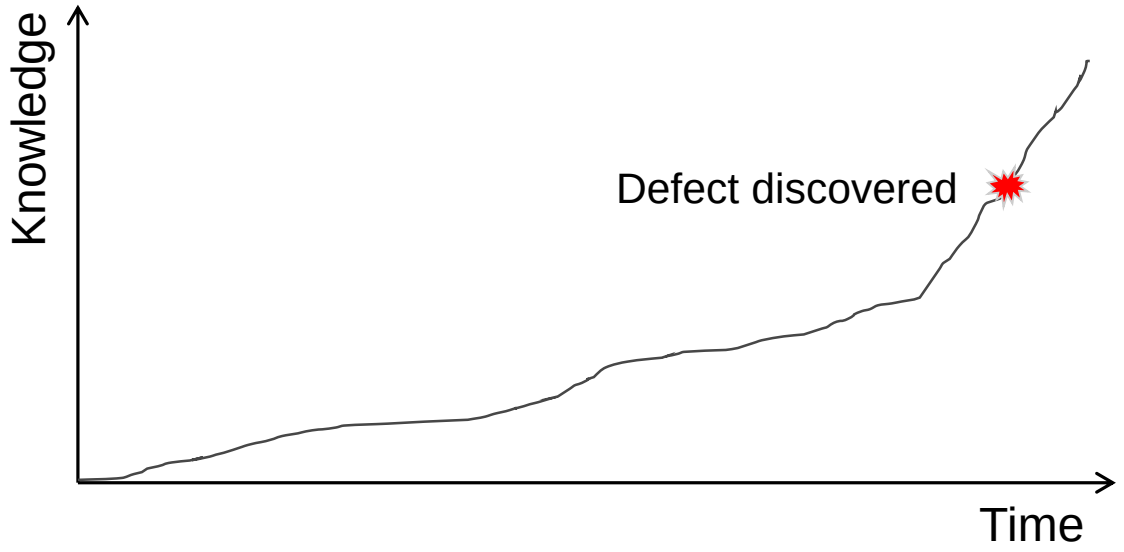
LEARNINGS FROM SAAB MODELING DAY

Modelling can only be introduced by starting with a small scale, succeeding and then expanding

CORE IDEAS – BUILD KNOWLEDGE EARLY

Experiences from traditional development

- Many assumptions in early phases, slow growth of knowledge
- Verification of requirements do not ensure that the right product is built
- Validation cannot start until a lot of work has been done
- Complicated to communicate between system- and software engineers



Validation – the requirements are correct and the system is useful in its context

Verification – the system meets its requirements

CORE IDEAS – BUILD KNOWLEDGE EARLY

Early validation

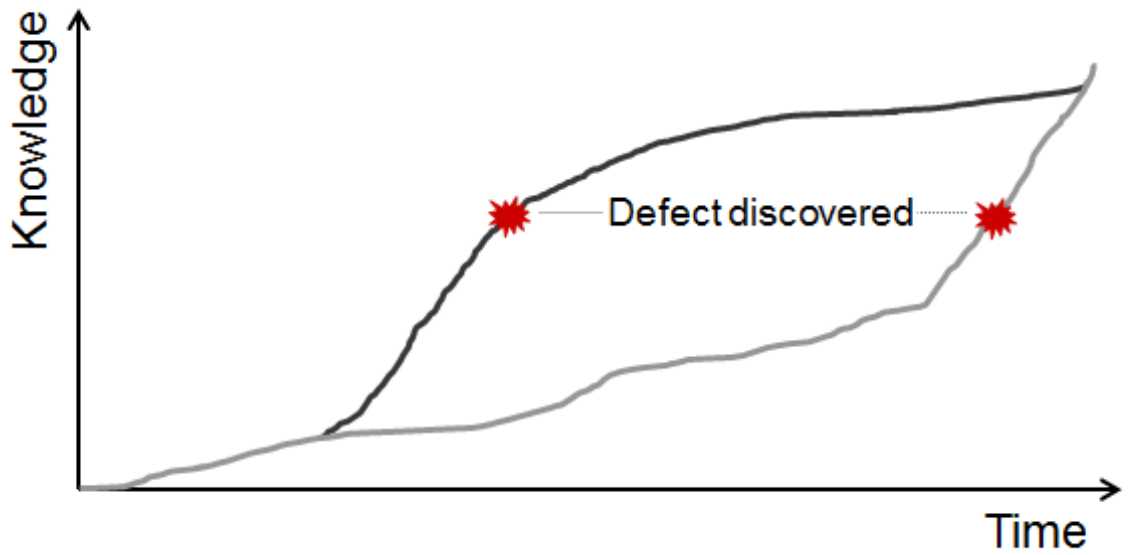
- Find problems earlier when they are less expensive to fix
- Build confidence that the realized system will work as expected

Executable models

- Validate intended function
- Build knowledge in early phases

Simulation of complete system on desktop

- Validate integration and presentation



MODELING@AERONATUTICS

GRIPEN, A GLOBAL SUCCESS STORY

Operated by air forces in five countries and the Empire Test Pilots' School in the UK.



Also purchased by



And selected by

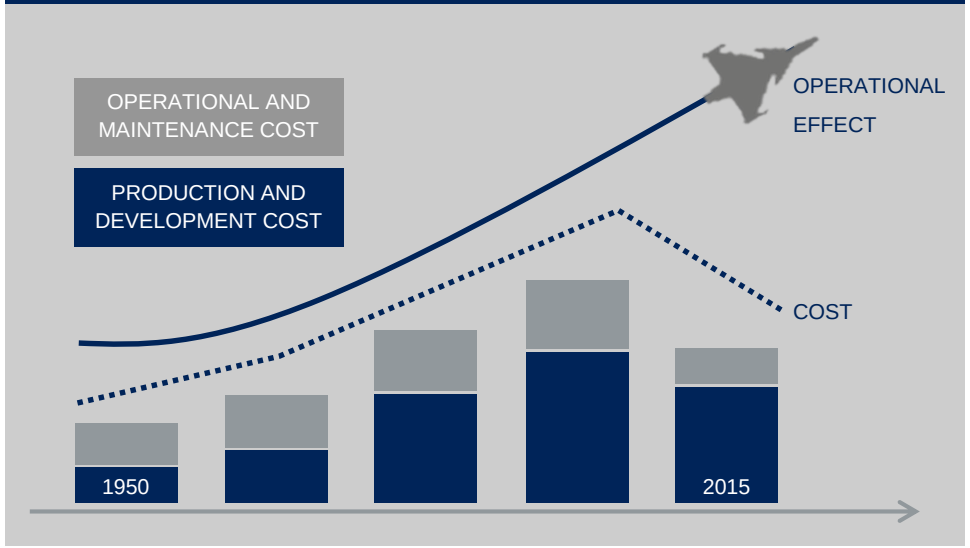


- High performance and outstanding availability
- Extremely agile and manoeuvrable, full multirole capabilities
- Small logistic footprint - cost-effective operations, upgrades and support
- Superior information sharing
- Very high weapon flexibility

Target: 300-450 aircraft sold over the next 20 years

This represents 10 percent of the accessible market

Gripen has won 50 percent of competed business opportunities



THE GRIPEN E VISION



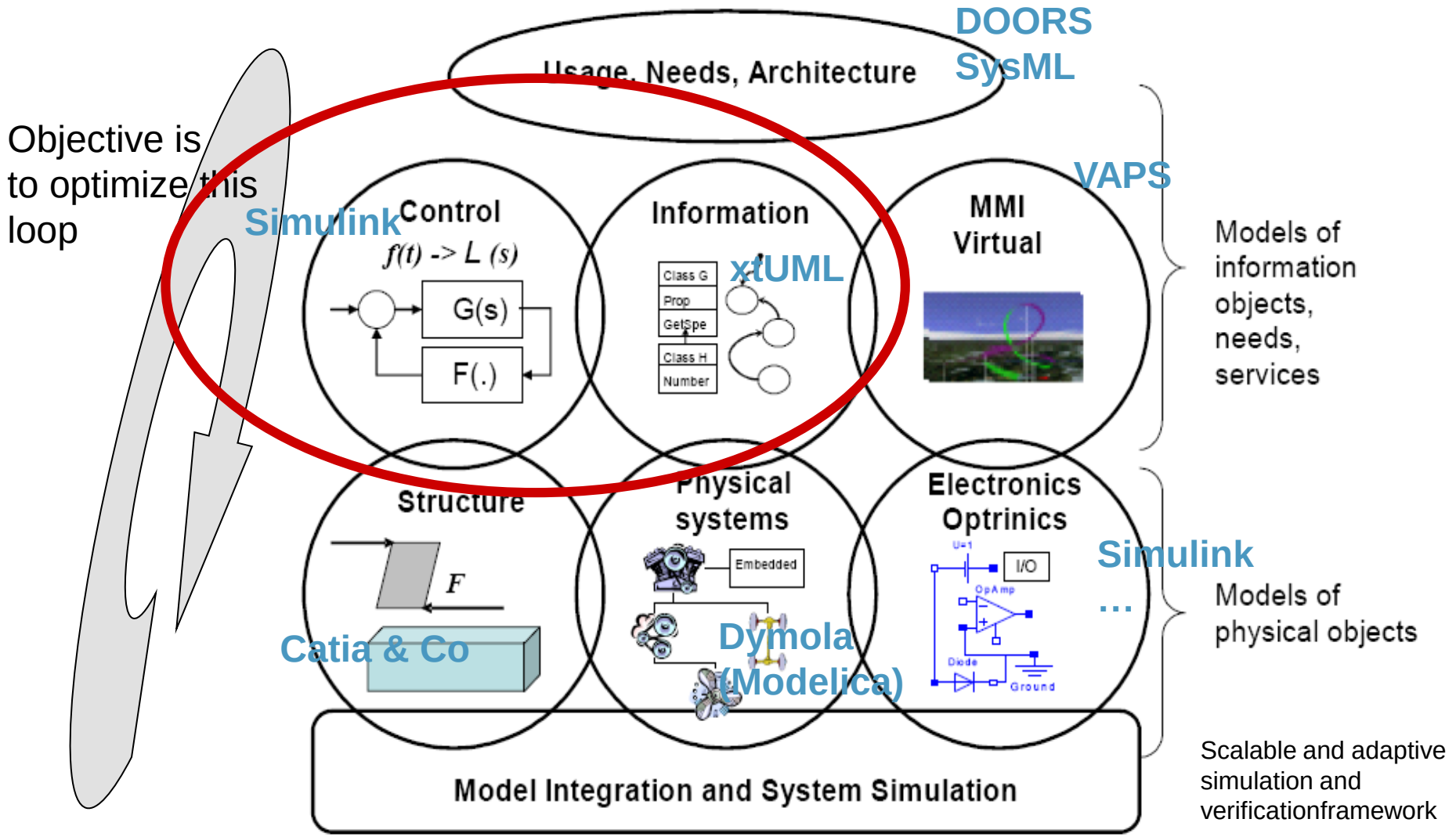
Vision from a development perspective:

- Possible to flight test new versions within days.
- Typical development cycle reduced in time and cost with 30-50%
- Possible to deliver new variants with minimum of additional testing.

How:

- Systems architecture supporting separation (criticality, options, development, …)
- MBSE for early validation
- A/W – optimized documentation

MODELLING EXPERIENCE



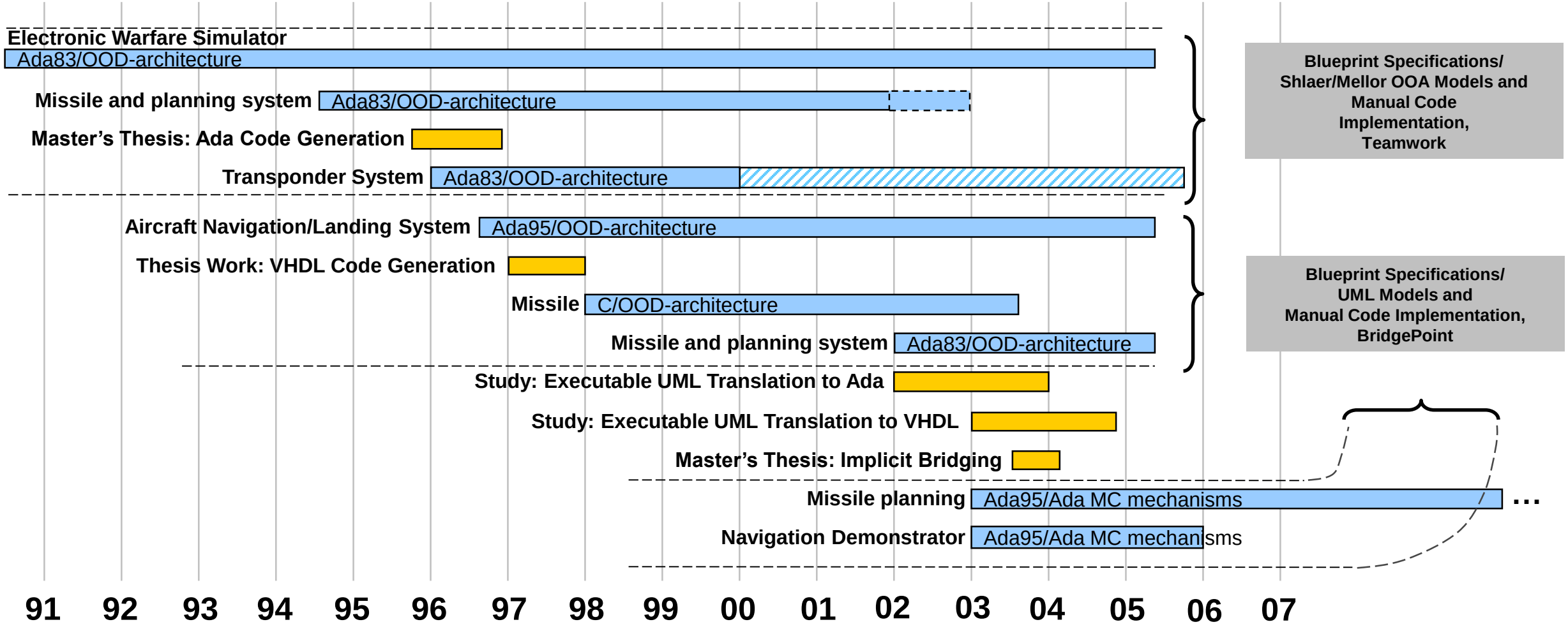
CORE IDEA – CONSISTENT INFORMATION

- ✓ Usage of auto-generated code approved
- ✓ Software Development Plan for Gripen E includes MBSE
- ✓ Manuals and instructions how to use model information as design data
- ✓ Verification tools to automate inspections
- ✓ Inspection effort moved from code to model

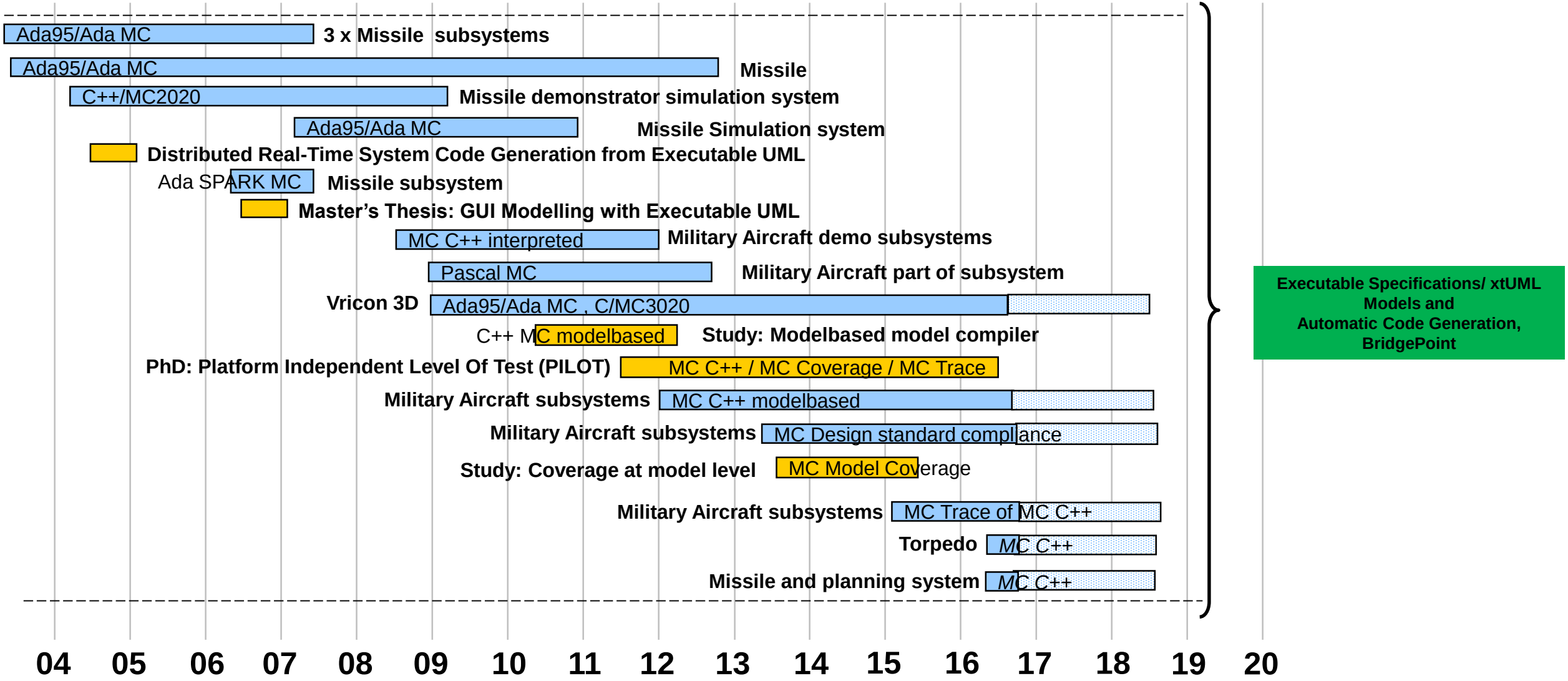
Vision

- Use models where suitable do express the design (on different levels)
- For software models, work in models with no manual inspection of the code

HISTORICAL XTUML/MDA DEVELOPMENT

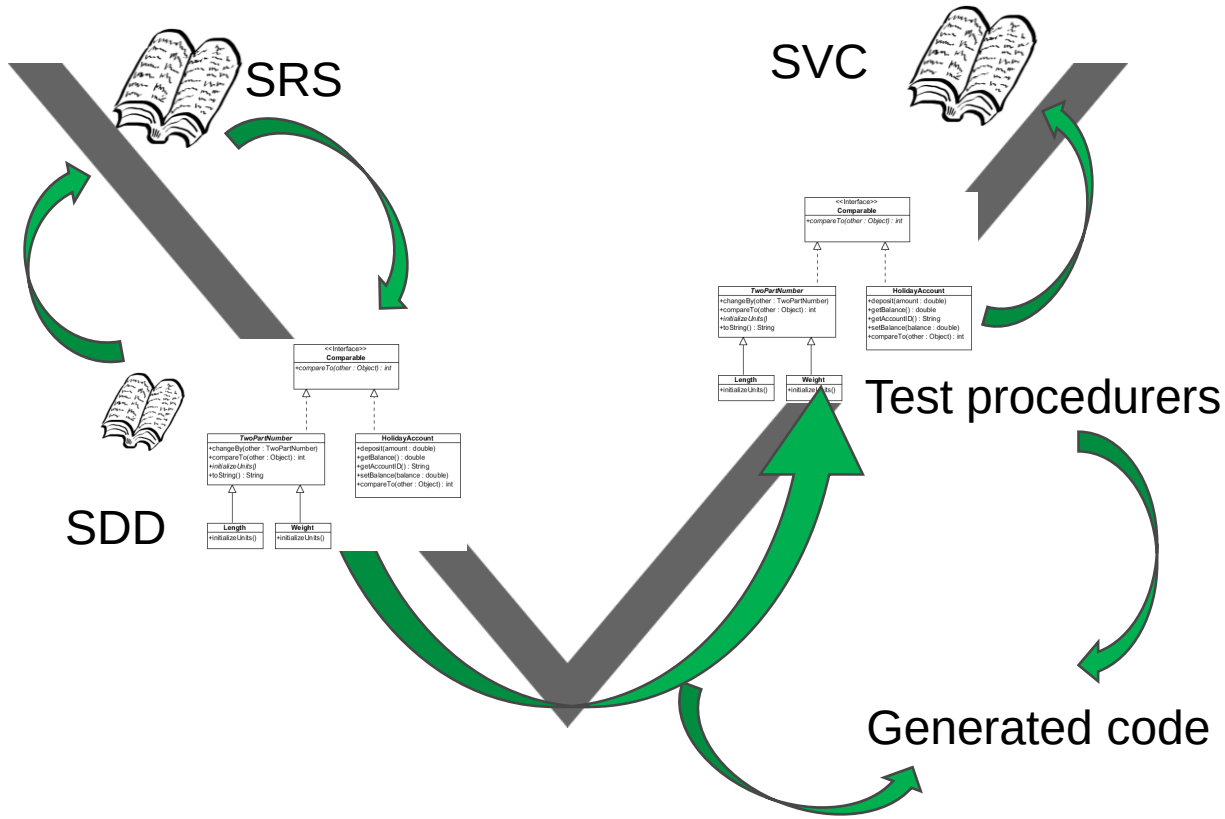


HISTORICAL XTUML/MDA DEVELOPMENT



USING XTUML IN GRIPEN E

- Can be used for all DAL-levels
  - Five software applications used xtUML at first flight of the new Gripen E
- No differences in high level requirements (SRS) and software verification cases (SVC) compared to hand code
- Most of the low level requirements (SDD) are found in model
  - Document mostly for tracing to requirements
- Test procedures
  - Hand coded or modelled
- Code generated from model



VISION FUTURE PRODUCTS

- The vision is that by the time of 2020 everyone working with the future products project shall use unambiguous models as the primary information source in order to perform responsive and cost effective SoS and system design and development to deliver products that fulfils the customers needs throughout the products life cycle.

Unambiguous modeling is here to stay at Saab

MODELING@DYNAMICS

CURRENT USAGE

- New xtUML defined “sw-product“
  - Users understand separation of functionality and architecture
  - xtUML components to define applications, domain definition is not the focus,
  - Using the xtUML components to define interfaces of Sw applications/components.
  - Generating definition for transport layer instantiation.
  - Generating bridge between transport layer and xtUML component (model and source code)
  - xtUML component may be model or used defined
  - Source code generation of modelled xtUML components
  - Models are very explicit

CURRENT USAGE

- Extisting xtUML defined “sw-product”
  - Porting from domain to component and interface usage
    - Keeping the domain concept as an outer component
    - Bridges is defined using components and interfaces. Local and system interfaces
    - Components in components are used to reuse “domains”
  - Both project specific and generic domains exist
  - Exisiting bridges have actions defined, impacts the portability
  - System view is now visible
  - MC issues

