Gianluca Monticone

Overview sulla proposta IBM Rational in merito al System Engineering

System Engineering: Smart Products

Agenda

- Introduction
- Integrated SE approach
- Value Offerings from IBM-Rational
- Benefits of a pre-configured process adoption
- An integrated Rational/Telelogic Solution

Agenda

- Introduction
- Integrated SE approach
- Value Offerings from IBM-Rational
- Benefits of a pre-configured process adoption
- An integrated Rational/Telelogic Solution

Top Reasons for missing cost, schedule and profit goals

Business Challenges

Product missed customer needs	46%
Late to market/missed demand	33%
Poor commercialization / promotion	26%
Product quality	24%
Pricing	23%
No clear product differentiation	19%

The CIO's Guide to the PERFECT Launch: Translating Innovation to Business Benefit, AMR Research, 2005

Engineering Opportunities

Improve communication and collaboration across disciplines	**71%**
Increase visibility into status of requirements	**49%**
Increase ability to predict system behavior prior to testing	**46%**
Implement or alter new product development processes for a multi-disciplinary approach	**43%**
Increase real time visibility of product Bill of Materials (BOM) throughout the development process	39%

Aberdeen Group, System Design: New Product Development for Mechatronics, Michelle Boucher, David Houlihan, January, 2008

What is Systems Engineering?

A *system* provides a set of services that are used by an enterprise to carr out a business purpose (i.e., mission) System components consist of hardware, software, data and workers*

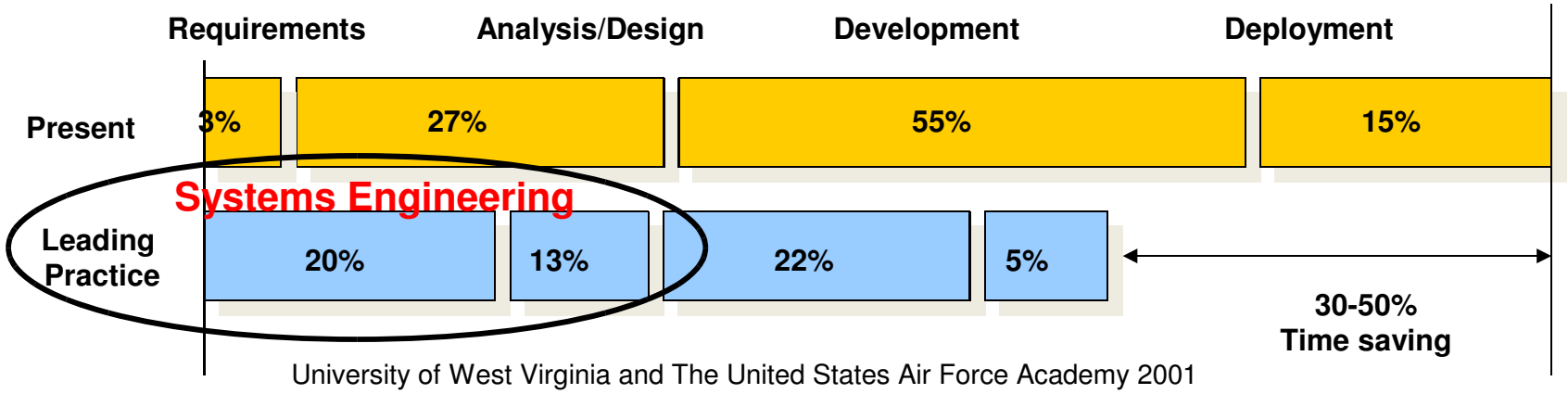
As defined by the International Council of Systems Engineering (INCOSE), systems engineering is:

"……Systems engineering integrates all the disciplines and specialty groups into a tear effort forming a structured development process that proceeds from concept to production to operation. Systems engineering considers both the business and the technical needs of all customers with the goal of providing a quality produ that meets the user needs."

Systems Engineering Challenges

- **Poor requirements engineering results in many failed projects**
  - Customer requirements not met
  - Capabilities not delivered due to cost/schedule overruns
- **Paper-based and manual processes hinder efficiency**
  - Without tool-based automation, many processes are slow and unwieldy
  - Impact analysis of changes can be time-consuming and expensive
- **Ineffective distribution of functionality across components**
  - Unable to juggle multiple factors to ensure that required functionality is achieved by the system
- **Hardware/software integration is often late**
  - Hardware and software may be first brought together at system test
  - Mistakes and rework are very expensive
- **Many organizations lack formalized practices**
  - Expertise is often in heads of “greybeards” and difficult to replace
  - Difficult to attract and retain talent without defined practices and career paths

Emphasis on Systems Engineering



- The gap in benefits between the highest and lowest IT spenders for computing power was only 4% without a good management system.
- What we need is a good governance system
  - Productivity improvement is 25%
  - Capital improvement is 70%

Source: Stephen J. Dorgan and John J. Dowdy - The McKinsey Quarterly, 2004 Number 4

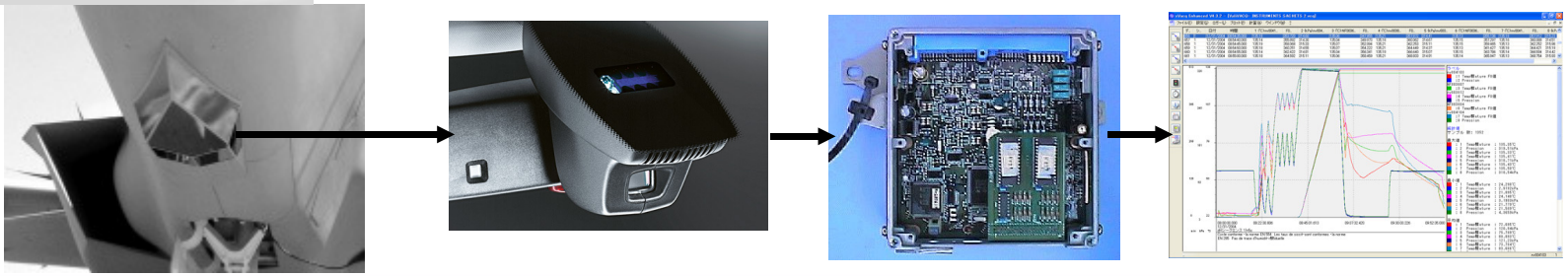
Example of a System-Level Design Failure

Problem at stake: Provide automation through sensing of information

- Particular instance: Rain Sensing Wiper
- Real-life example, illustrating system-level complexity i products

Constraints

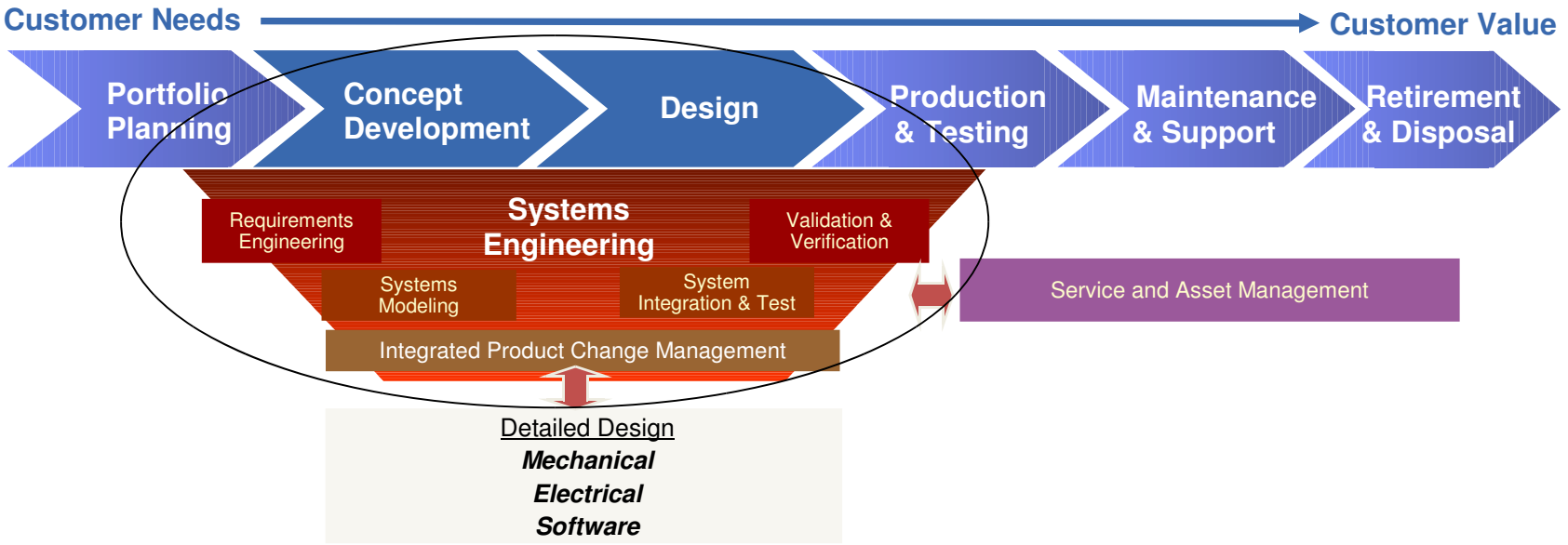
- Mechatronic development: Electrical, Mechanical and software
- World-wide distributed supply chain

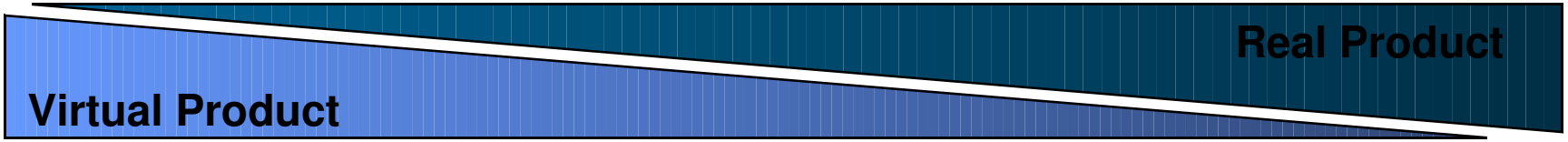


Agenda

- Introduction
- Integrated SE approach
- Value Offerings from IBM-Rational
- Benefits of a pre-configured process adoption
- An integrated Rational/Telelogic Solution

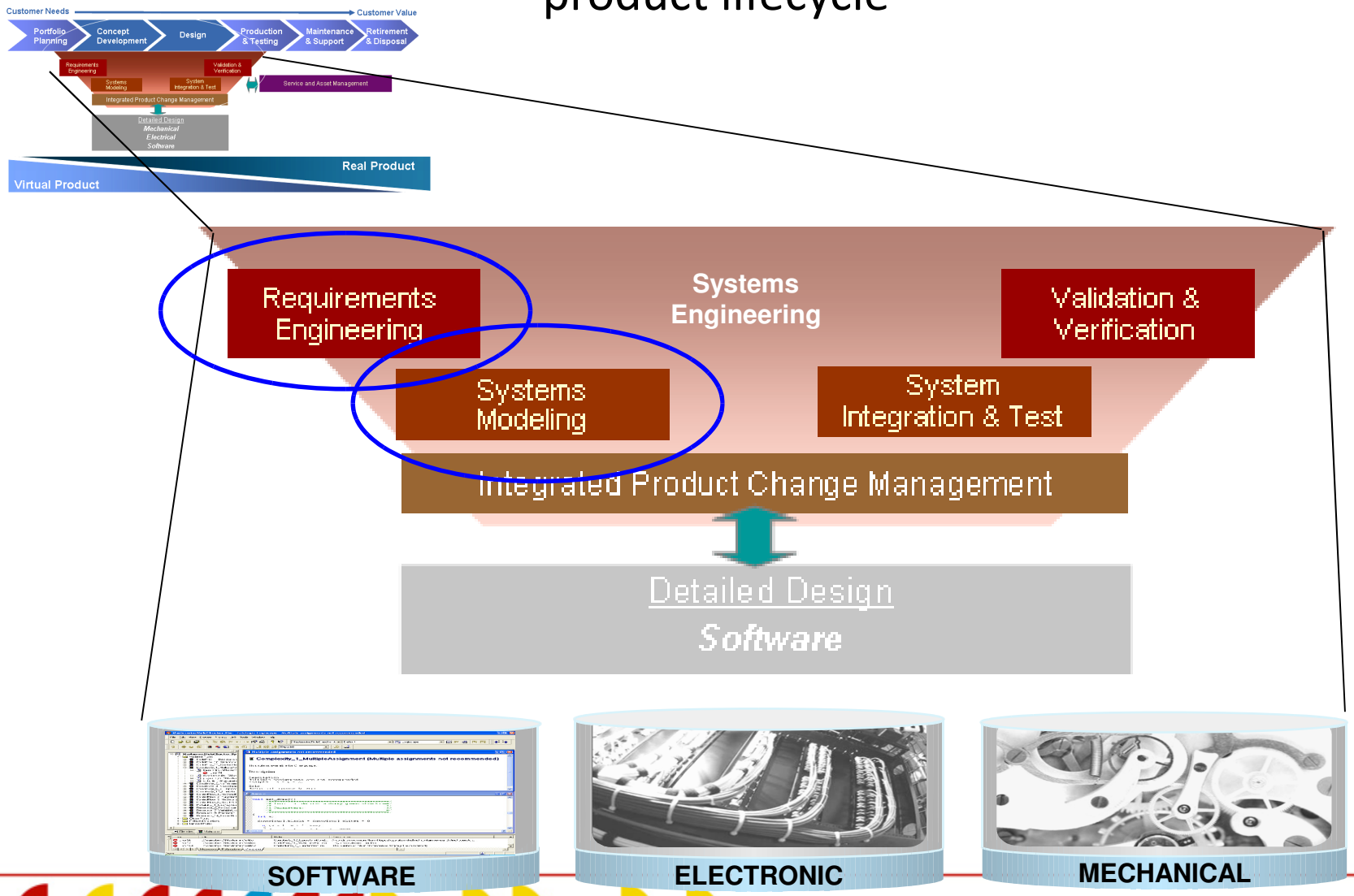
Systems engineering is a key part of product development





Early in the lifecycle, 'virtual product' design is used to make better decisions to ensure financial viability and technical feasibility of new products, resulting in increased quality, reduced time to market and lower costs

Systems Engineering unites engineering disciplines throughout the product lifecycle



There is a Significant Impact to the Business

As a Result or a Poor Requirements Engineering Process

Requirements Rework

- Errors, late detected in the Maintenance phase can cost up to **200 times** more than detected early in Requirement Analysis phase[1]
- **More than 40%** of development budget can be consumed by poor requirements[2]

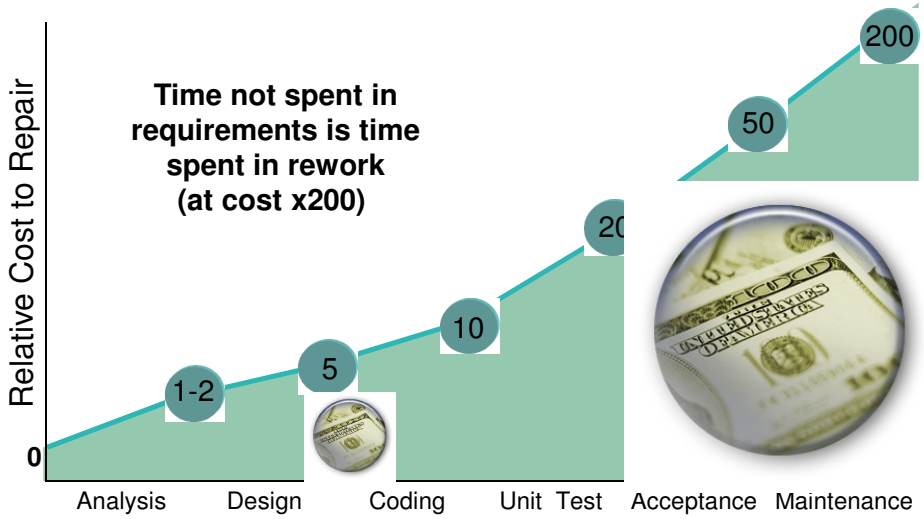
Project Impacts

- **41%** of projects fail to deliver the expected business value and ROI[3]
- **49%** of projects overrun original estimates[3]
- **28%** of projects on time and on budget[4]

Requirements Delays

- Being late to market by 6 months or more will cost organizations 33% of the 5-year ROI**5**

Requirements issues drive excessive rework, delays, poor quality, and project failures



Stage in which Requirements Error Is Discovered

"Our research indicates 80-plus percent of software development failures result directly from poor requirements gathering, management, and analysis."

IDC, November 2007

Sources: 1) Leffingwell & Widrig, "Managing Software Requirements," Addison Wesley, 1999 2) IAG Consulting, 2008 3) Dynamic Market Limited, 2007 4) Standish Group, 2001 5) Don Reinertsen, McKinsey, 1983

Requirements Engineering Process & Tool Enforcement ROI

Quantifiable savings

- **Advanced Tomahawk Weapon Control System – incremental development lifecycle**

Improvements	Before	After
Requirements Volatility -		
@ Preliminary Design Review	72%	48%
@ Final Design Review	33%	17%
Requirements Changes Implemented -		
Changes accepted	98%	16%
Changes rejected	2%	84%
Testing Time -		
Integration	9 weeks	4 weeks
System	13 weeks	6 weeks
User Acceptance	22 weeks	10 weeks
Defects found after production	728	165
Software Requirements Specification	10 days	2 days
Production Time		

Impact analysis could not be done, so most changes were accepted

Impact analysis took up to 2 days, now only a matter of minutes

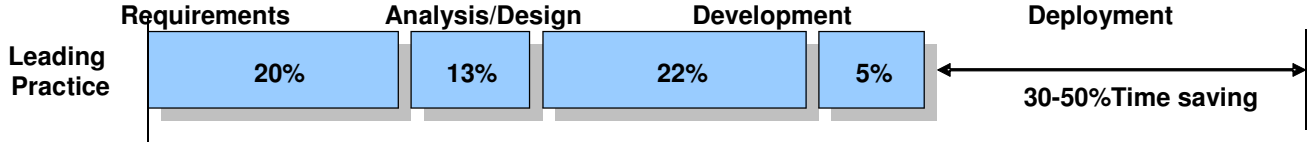
Key Principles of Systems Development

- **Model system complexity**
  - Systems engineers focus on big picture to ensure requirements can be satisfied
  - Dependency linkages throughout design improve impact analysis and traceability
  - Unite mechanical, electrical and software engineering
- **Simulate systems behavior with executable models**
  - Understand cross-disciplinary relationships earlier in development
  - View impact of proposed changes
- **Perform architectural analysis early in the design process**
  - Ensure business objectives are satisfied by the architecture
  - Evaluate both logical and physical architectures

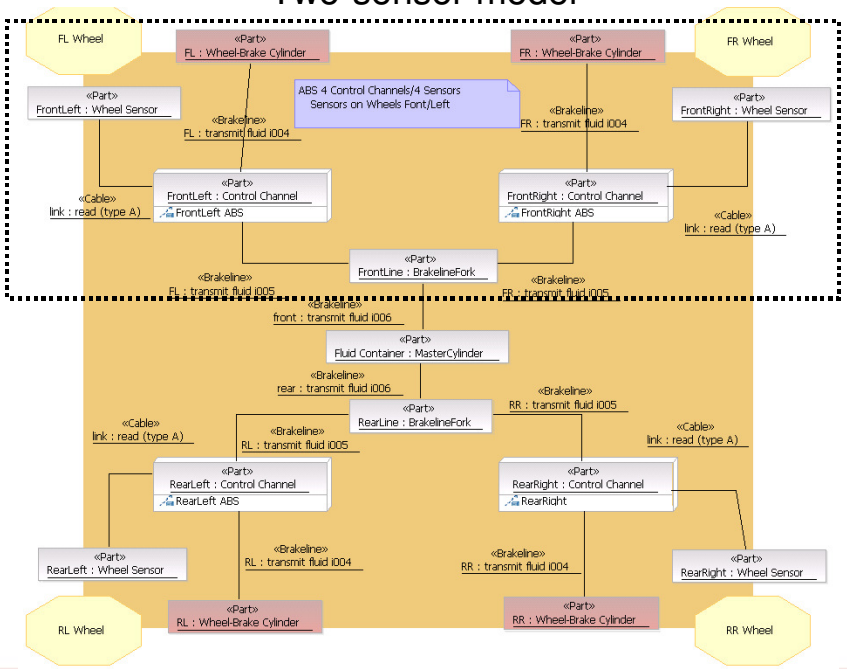
System-Level Design and Architecting: Business Decision

Communicate across teams and make business decision

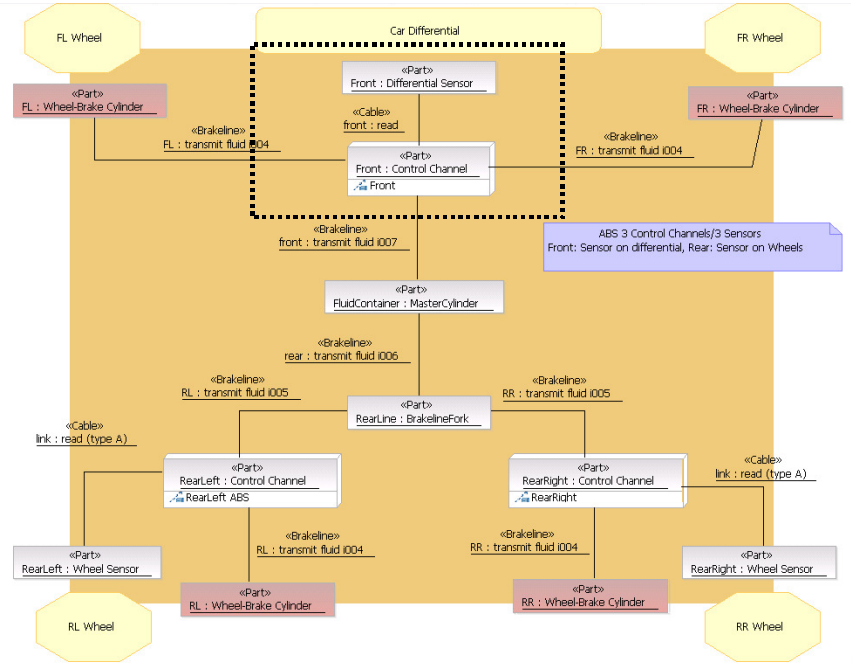
- Validate compliance with cross-architectural requirements
- Enable early decision-making on system configurations, perform trade-off analysis



Two-sensor model



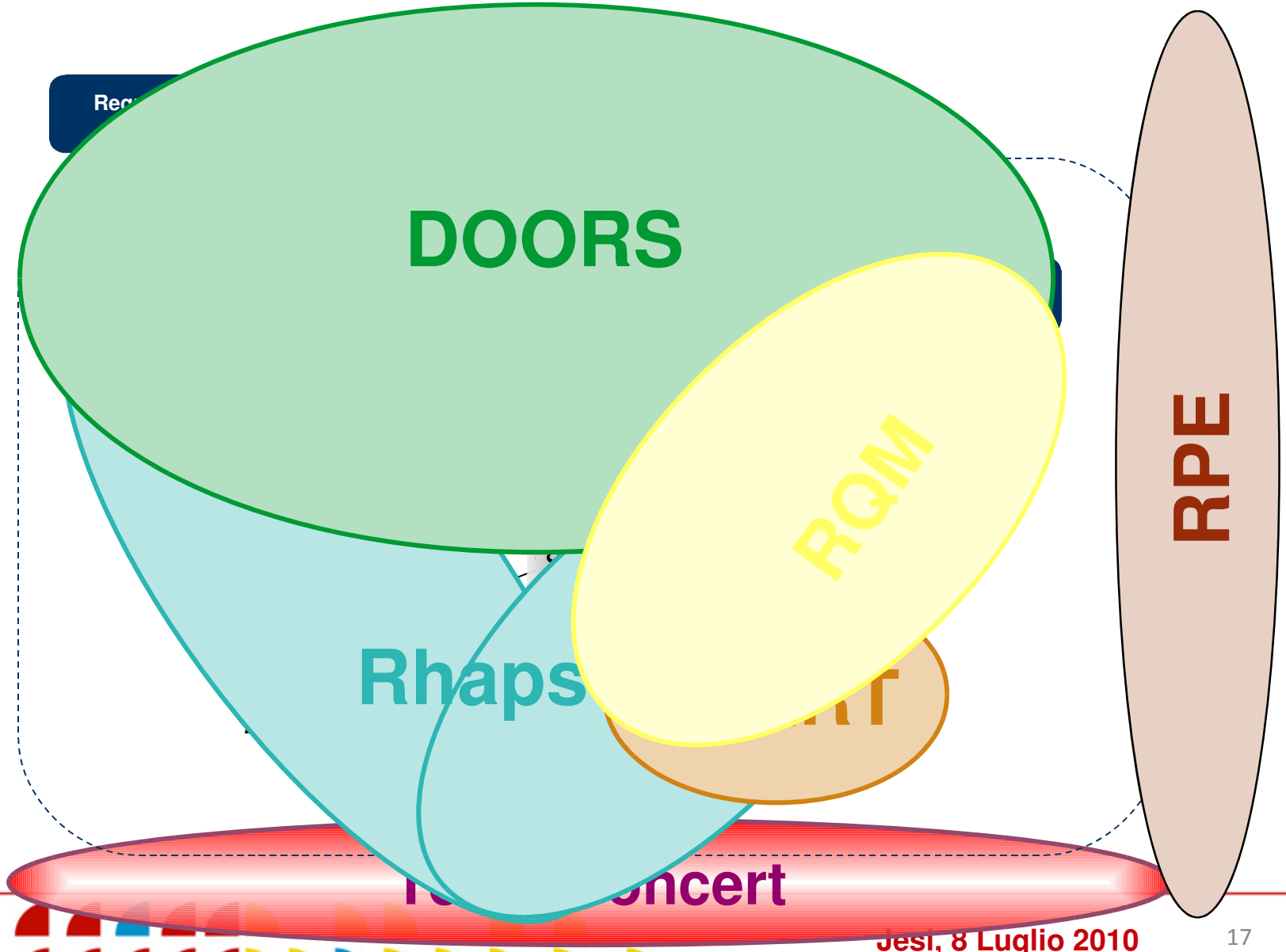
One-sensor model



Agenda

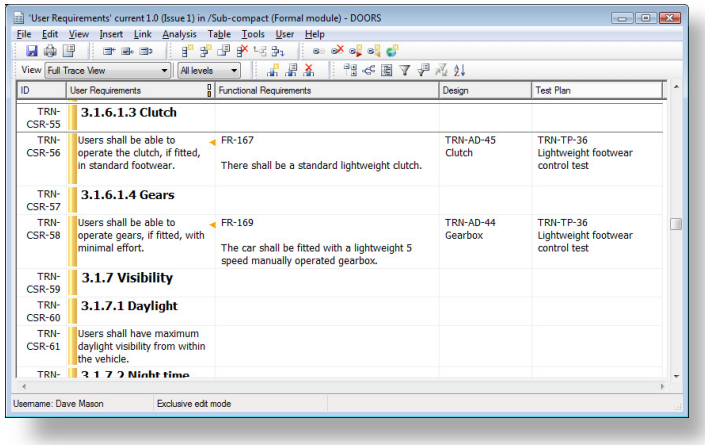
- Introduction
- Integrated SE approach
- Value Offerings from IBM-Rational
- Benefits of a pre-configured process adoption
- An integrated Rational/Telelogic Solution

The Classical System Engineering Workflow Where our Tools Play

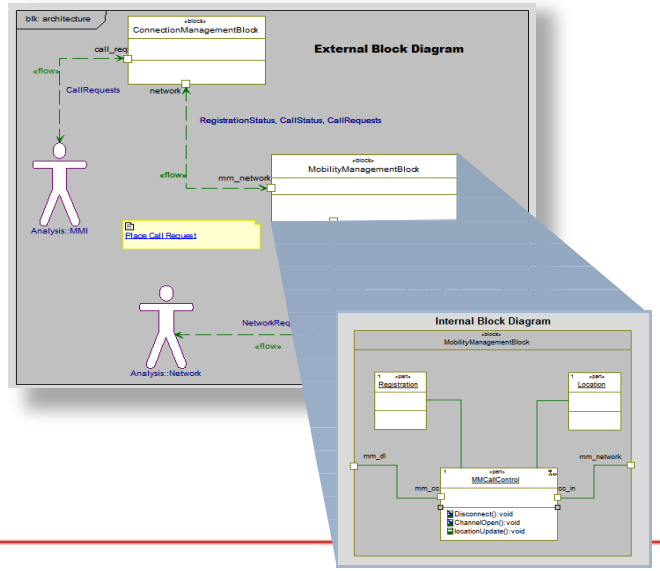


IBM Rational DOORS and Rhapsody

- **IBM Rational DOORS**
  - Requirements Engineering is a cornerstone of Systems Engineering
  - Close integration between IBM Rational DOORS and Rhapsody enhances Requirements Engineering capabilities



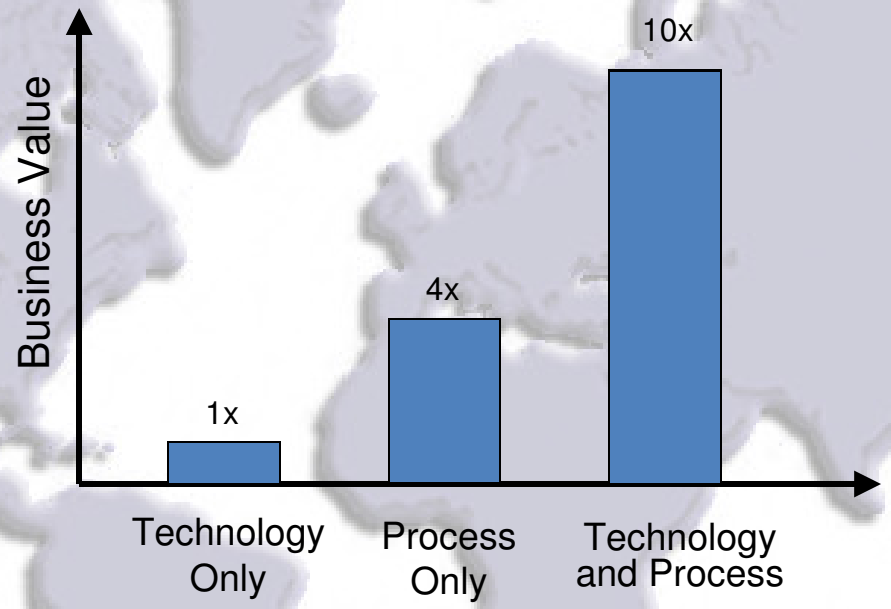
- **IBM Rational Rhapsody**
  - Execution of models facilitates early validation of design; catches problems earlier in lifecycle to greatly reduce cost of errors
  - Full UML 2.1 and SysML support
  - DoDAF reporting capabilities
  - Integrated with Harmony/SE for integrated process and tool guidance



Agenda

- Introduction
- Integrated SE approach
- Value Offerings from IBM-Rational
- Benefits of a pre-configured process adoption
- An integrated Rational/Telelogic Solution

Combination of Technology and Process Results in 10x Business Value



Source: London School of Economics – McKinsey Survey

Rational Harmony

- Library of re-useable best practices
  - Standard meta-model (OMG SPEM 2.0)
  - Standard tool (EPF Composer/RMC)

- Pre-configured processes include:
  - Harmony/ITSW
    - IT Software
  - **Harmony/SE**
    - **Systems Engineering**
  - Harmony/ESW
    - Embedded Real-time Software

Agenda

- Introduction
- Traceability Across Enterprise
- Integrated SE approach
- Value Offerings from IBM-Rational
- Benefits of a pre-configured process adoption
- An integrated Rational/Telelogic Solution

Envisioning a platform that can transform software delivery

Jazz is a project and platform for *transforming how people work together* to deliver greater value and performance from their software investments.

- robust, extensible and scalable
- globally distributed, fluid & dynamic
- community-based & open at Jazz.net

Collaborate in Context

- Enable team transparency of "who, what, when, why"
- Build team cohesion and presence
- Automate hand-offs – so nothing falls through the cracks

Right-size Governance

- Automate team workflow improving productivity
- Automate data collection eliminating administrative overhead
- Real time reporting and alerts reduces project risk

Day One Productivity

- Dynamic provisioning of projects and teams
- Real-time iteration planning and workload balancing
- Unify teams with tools choice

Dynamic integration of people, process and projects across the lifecycle