

WHY AND WHERE DO WE NEED EXECUTABLE MODELING?

Presented by Francis Bordeleau
francis.bordeleau@ericsson.com



OUTLINE

- CONTEXT
 - PERSONAL BACKGROUND
 - ERICSSON
- SUPPORT FOR EXECUTABLE MODELING
- WHERE CAN EXECUTABLE MODELING BRING VALUE
 - SOFTWARE DESIGN
 - OTHER ASPECTS
- KEY CHALLENGES
- SUMMARY

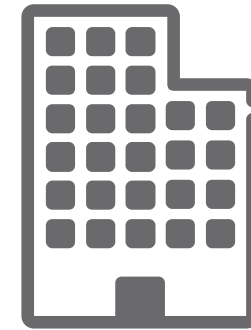
PERSONAL BACKGROUND



RESEARCH/
ACADEMIA
10 YEARS



CEO OF SME
10 YEARS



PRODUCT MANAGER
2.5 YEARS



TEACHING



CONSULTING



TELECOM



SDR



DEFENCE
AEROSPACE

ERICSSON IN NUMBERS



40% OF ALL
TRAFFIC



2.5 BILLION



180 COUNTRIES



110,000
EMPLOYEES



5TH LARGEST



50 BILLION

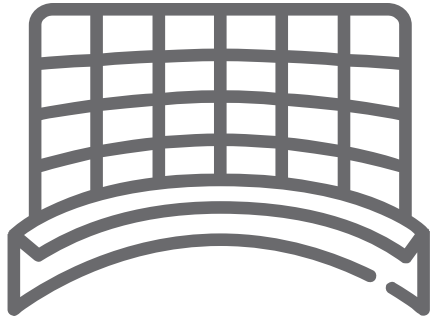
DRIVING PRINCIPLES



- › Modeling is not an end-goal, it is a means to an end
System/application deployment is the end-goal!
- › Our main goal: Improve processes/tools to make development more efficient
 - We need to build system/application faster, cheaper, with increased quality, and reduced risks
 - We need to develop scalable methods to allow addressing increasing system complexity
- › Our work must be based on strong business cases
 - Without business cases, there is no software engineering

We need to be pragmatic ... not dogmatic!

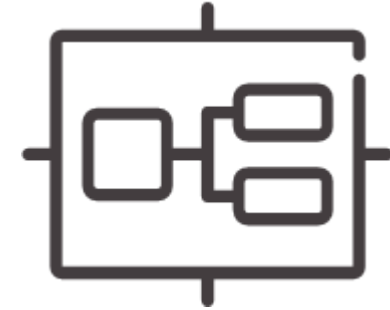
MODELING IN ERICSSON



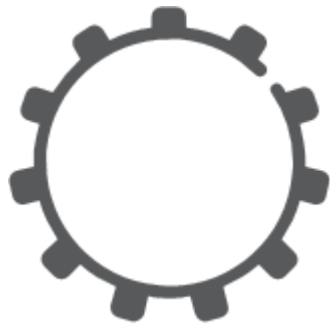
INFORMATION
MODELING



NETWORK
MODELING



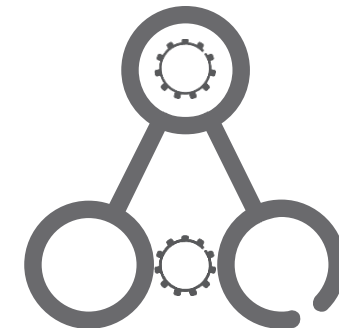
SYSTEM/
SOFTWARE
MODELING



BUSINESS
PROCESS
MODELING



REAL-TIME
MODELING



EXECUTABLE
MODELS

WHY MODELING AT ERICSSON?



› Excellent development efficiency

- Quality
- Lead Time
- Flexibility



› Single source of information supporting different views

› Abstraction

› Improved information flow between the different development phases

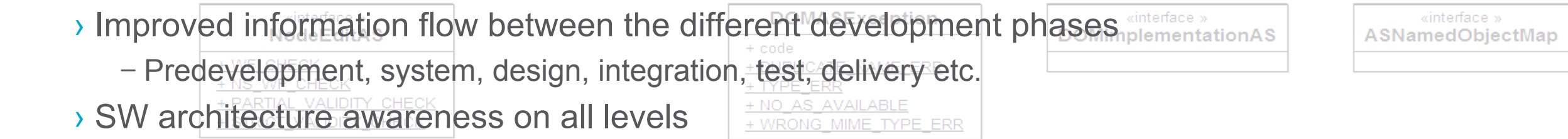
- Predevelopment, system, design, integration, test, delivery etc.

› SW architecture awareness on all levels

› **Early design execution / simulation / validation**

› **Extensive use of automation, validation and generation**

› **Supports simplified processes (e.g. Agile)**



DIFFERENT SYSTEM ASPECTS



Need to consider different system aspects and their respective needs

- › Requirements modeling
- › Software design -- high-level design and detailed design
- › System modeling
- › Functional modeling
- › Information modeling
- › Network modeling
- › Business process modeling

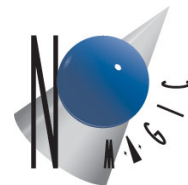
Where do we need executable modeling?

What type of executable modeling do we need?

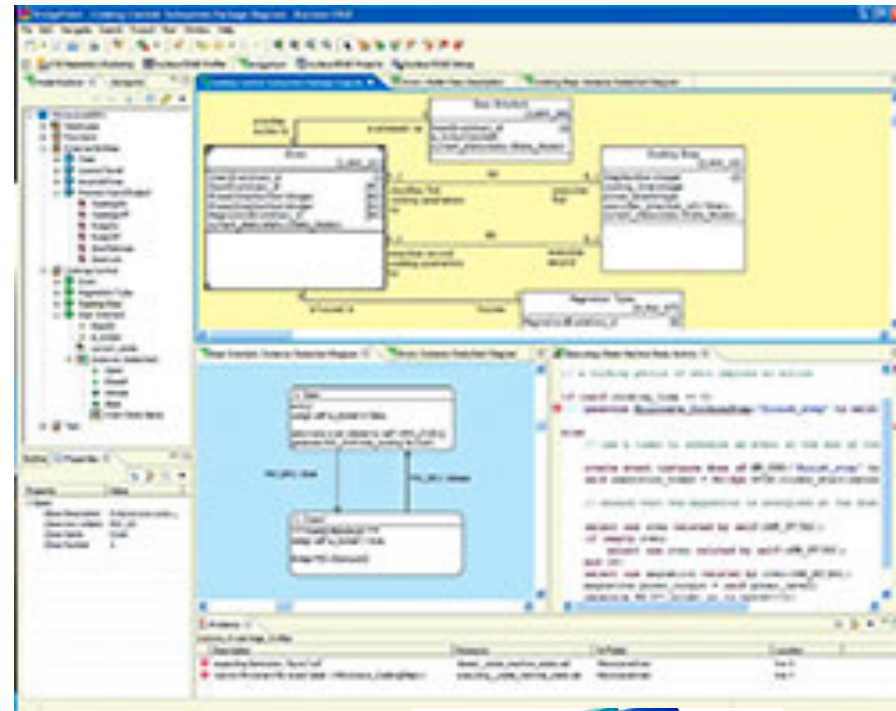
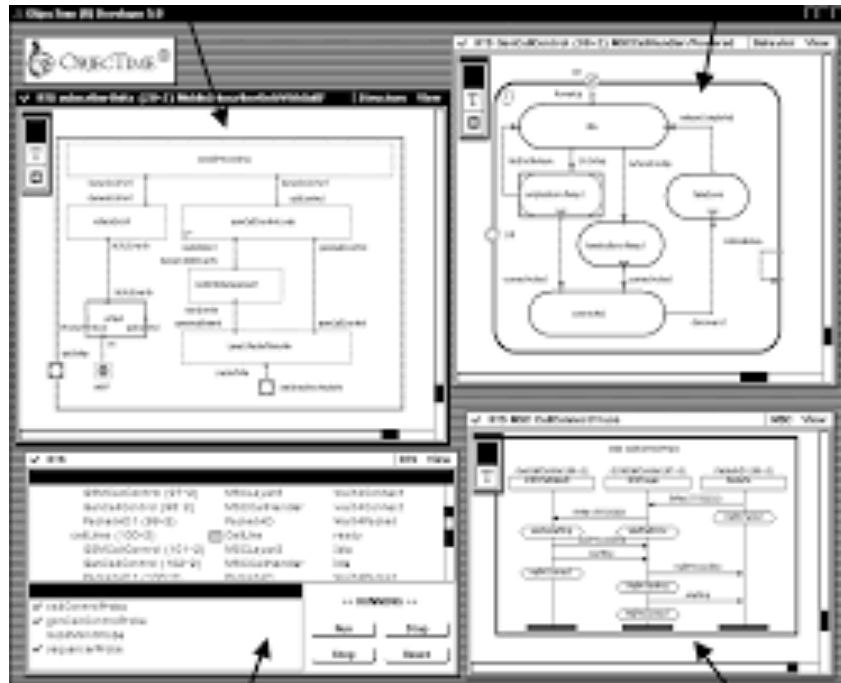


SUPPORT FOR EXECUTABLE MODELING

- › The idea of executable modeling is not new
- › It is at the core of the MBE vision
- › Over the years, different modeling tools have provided different types of support
 - This list is far from being exhaustive.
 - F. Ciccozzi, I. Malavolta, B. Selic. Execution of UML Models. Work in progress



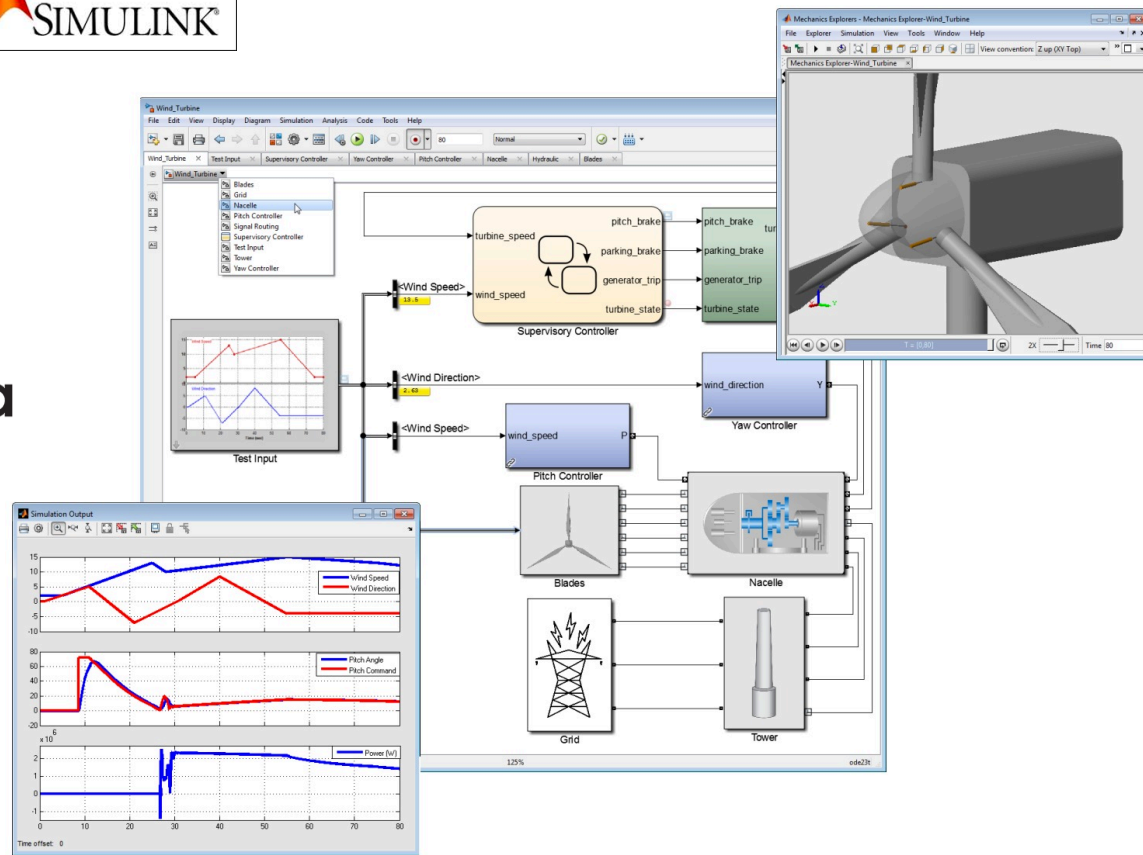
SOFTWARE DESIGN MODELING



FUNCTIONAL MODELING

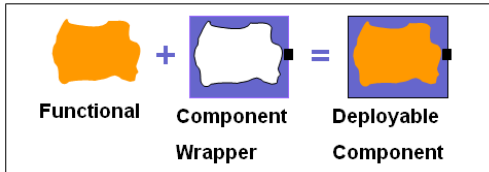


OpenModelica



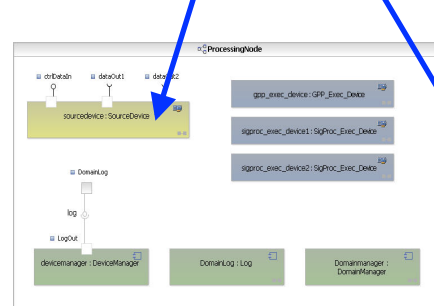
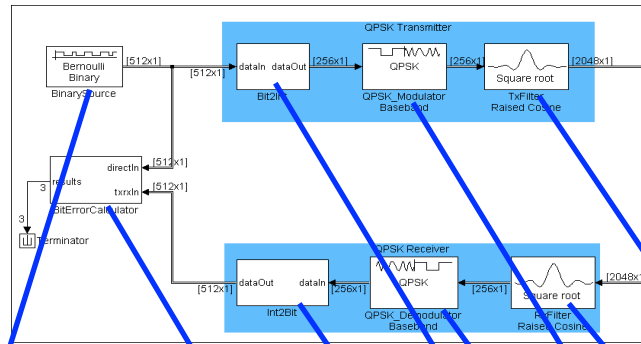
- Investigation of design alternatives
- Early error detection
- Rapid prototyping
- Analysis of emerging behavior

SOFTWARE DEFINED RADIO

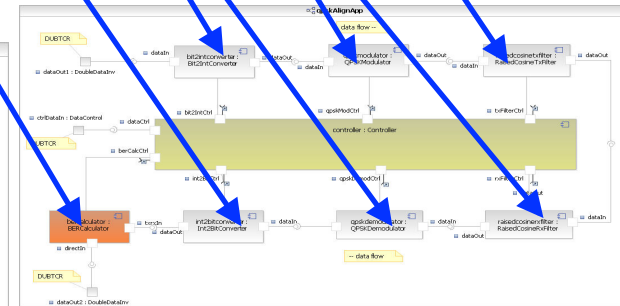


- Investigation of design alternatives
- Early error detection
- Rapid prototyping

Functional Specification



Platform



Component-Based Application

WHY DO WE NEED EXECUTABLE MODELING



› Validation, verification, testing

- Some of it can be done without execution, but ...

› Discovery of emerging behavior

- In complex system, the overall behavior can't be understood by simply understanding the individual components

› Execution is cool!

- I want to execute my model

- This is very important

- Execution makes modeling much more fun

EXECUTABLE MODELING -- REQUIREMENTS



› Modeling language with executable semantics

- Precise semantics is not sufficient, must also be executable

- Precise vs detailed

 - › Models can be incomplete, but they need to be unambiguous and they need to contain sufficient level of details to enable execution

- For analysis and design, the ability to execute incomplete models is fundamental

› Behavior description language – action language

- Model execution require behavior specification

› Execution

- Generation or interpretation (virtual machine)

› Tooling support

- We need complete tooling support

- Both observability and controllability are required – Model-based tracing and debugging

- Tooling must support execution both “on host” and “on target”

MODEL EXECUTION



› Generation vs Model Interpretation?

- Both have their respective pros and cons
- Depends on development aspects and contexts

› Jordi Cabot, Executable models vs code-generation vs model interpretation,
<http://modeling-languages.com/executable-models-vs-code-generation-vs-model-interpretation-2/>

› Johan Den Haan, Model Driven Development: Generation or Model Interpretation?,
<http://www.theenterpriseearchitect.eu/blog/2010/06/28/model-driven-development-code-generation-or-model-interpretation>

COST BENEFIT ANALYSIS



Costs

- ›Imposes modeling constraints -- model must be precise
- ›Requires sufficient level of details -- model must be detailed enough to be executable
- ›Requires behavior specification
 - Additional development effort
- ›May require learning a new behavior specification language
- ›Requires proper development environment/tooling

Benefits

- ›Ability to execute the model to discover emerging behavior
- ›Early validation and error detection
- ›Automated model transformation (code generation) can be used to optimize code in specific programming languages
- ›The model is the design/implementation
- ›... and it makes modeling much more fun!

EXECUTABLE MODELING AND AGILE DEV -- SW DESIGN ASPECT



›Key questions

- Does “Modeling = Heavy Process”?

›Main issue with existing MBE tools

- Lack of proper support for detailed design “inner loop”

- Led to a wide spread perception that MBE and Agile Development are incompatible

›Can executable modeling contribute to make MBE more agile?

- Certainly!

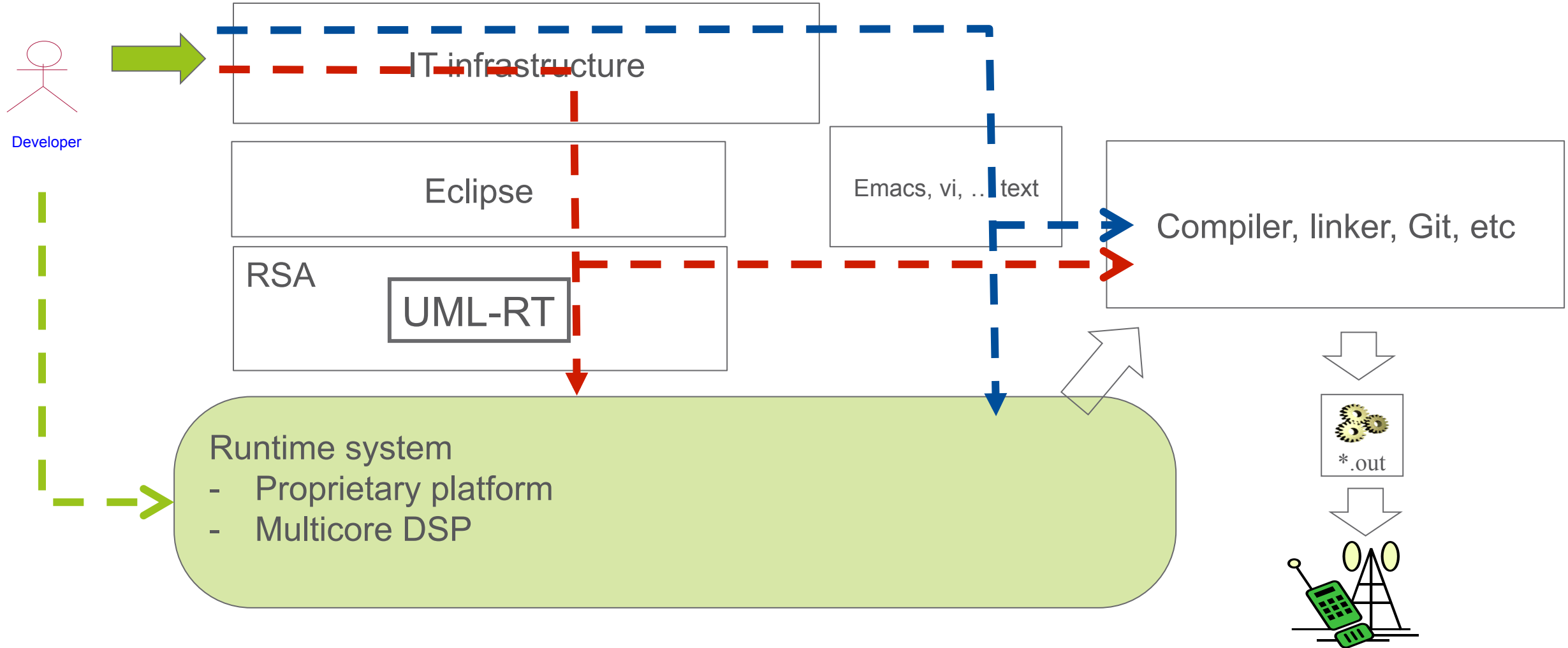
- But proper tooling support is required

›In many contexts, what is most needed is better integration with coding IDE

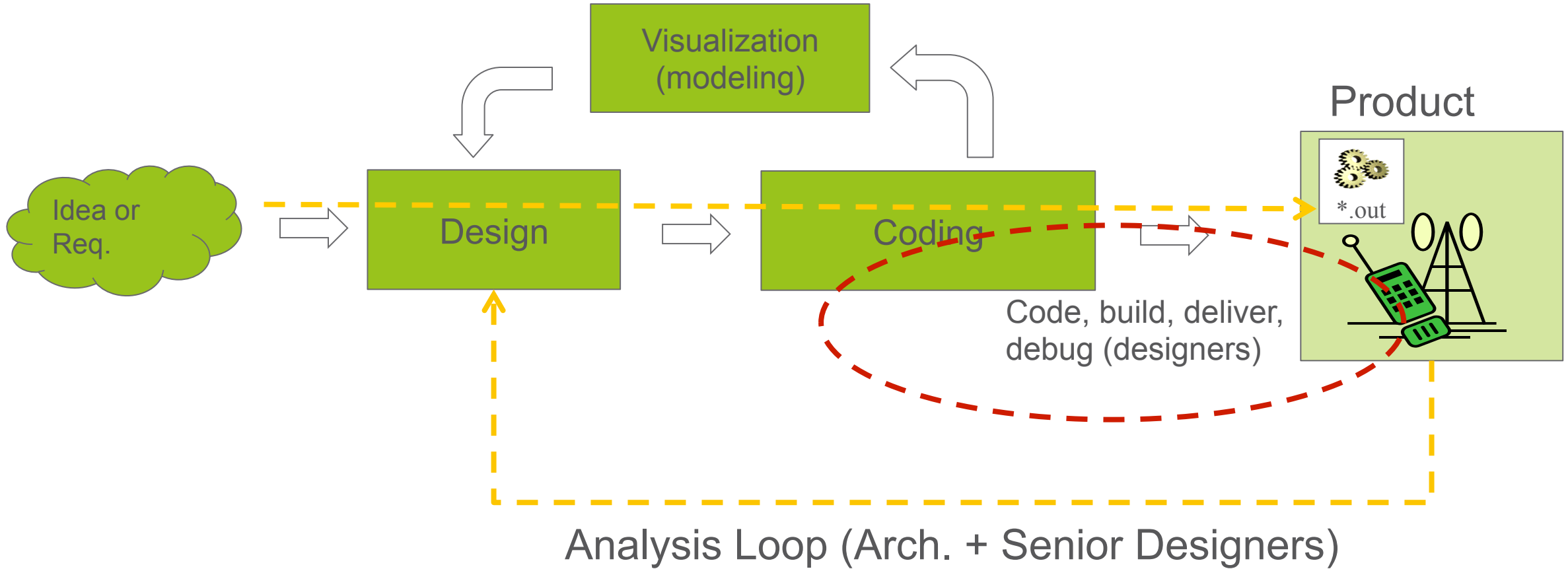
- Model-based tracing and debug

- Integration with legacy code and libraries

ERICSSON ISSUE IN BASEBAND DEVELOPMENT



THE SACRED INNER LOOP...



OTHER MODELING ASPECTS



- › What about other system development aspects?
 - Requirements modeling, system modeling, information modeling, network modeling, business process modeling?
- › Executable modeling can bring key value to aspects like system modeling, network modeling, business process modeling
 - Behavior discovery
 - Early validation and error detection
- › What about requirements modeling and information modeling?

KEY CHALLENGES FOR ADOPTION



- › Tool support
 - We need a complete model-based IDE, i.e. editor, compiler, debugger, etc
- › Compatibility with existing modeling languages
 - UML, UML-RT, xtUML
 - Migration path – greenfield projects vs integration in exiting systems/applications
- › Role of executable modeling in overall MBE development process
 - Need to be clearly defined
- › MBE education and culture change
 - This takes time and can't usually be imposed
- › Community development -- This is a key success factor!
 - Need different types of expertise
 - End users, suppliers, research academia

SUMMARY



- › Executable modeling is a key part of the MBE vision
- › Key questions
 - Why and where do we need/want executable modeling?
 - What is the end goal of executable modeling?
 - Can executable modeling contribute to make MBE more agile?
 - Can executable modeling contribute to the broader adoption of MBE?

Executable Models can bring key benefits

... but needs and requirements are quite diverse depending on the modeling context and application domains!

ACKNOWLEDGEMENTS



- › Sébastien Gerard, CEA
- › Andreas Henriksson, Ericsson
- › Charles Rivet, Zeligsoft
- › Bran Selic, Malina



ERICSSON