Innovate2010

IBM開發者大會

# *Model-Driven Development for Systems Design and Software Development of Real-Time and Embedded Systems*


Ronald Cheng

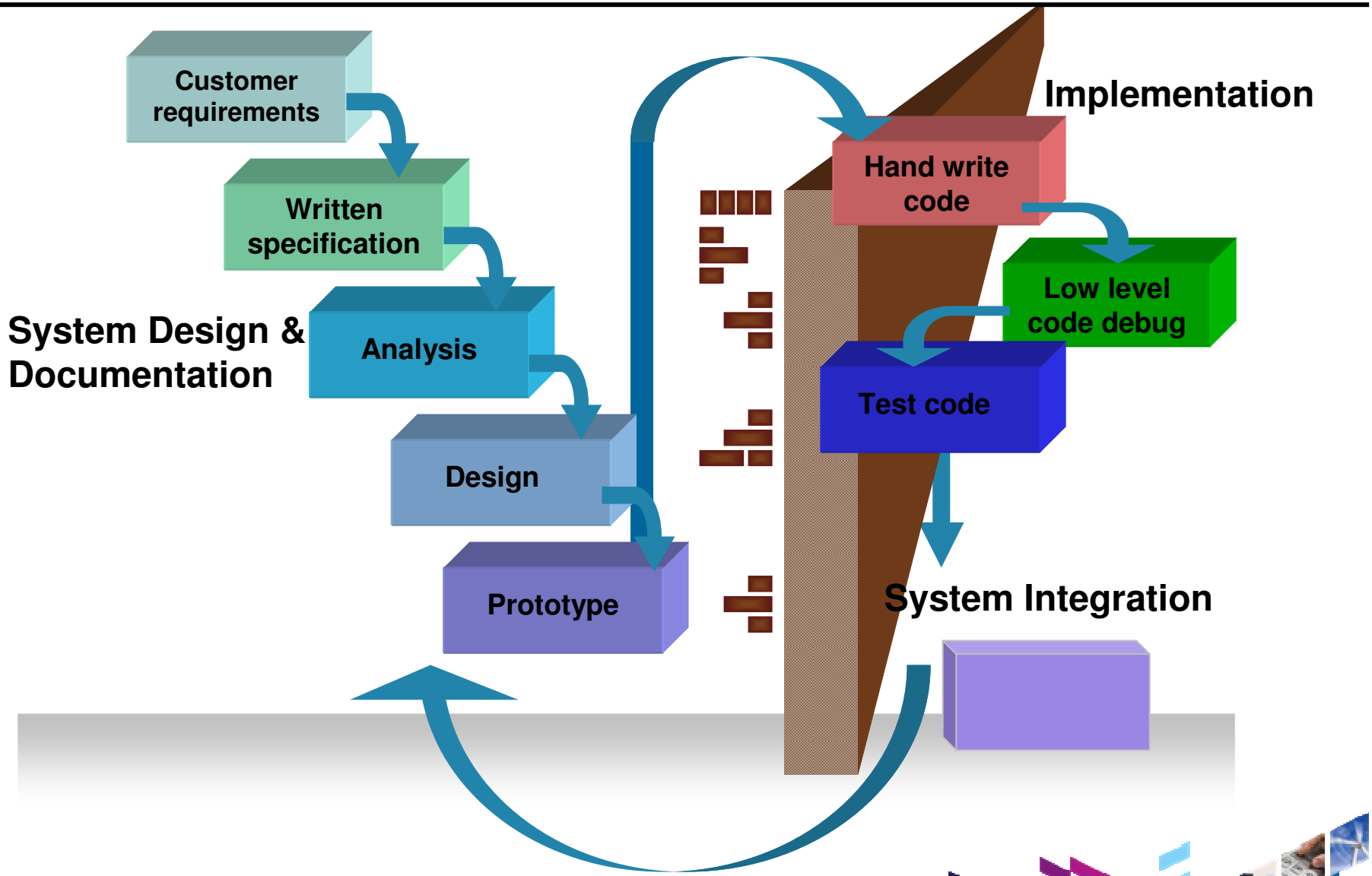
Products Specialist – MDD, Rational

ronaldc@hk1.ibm.com

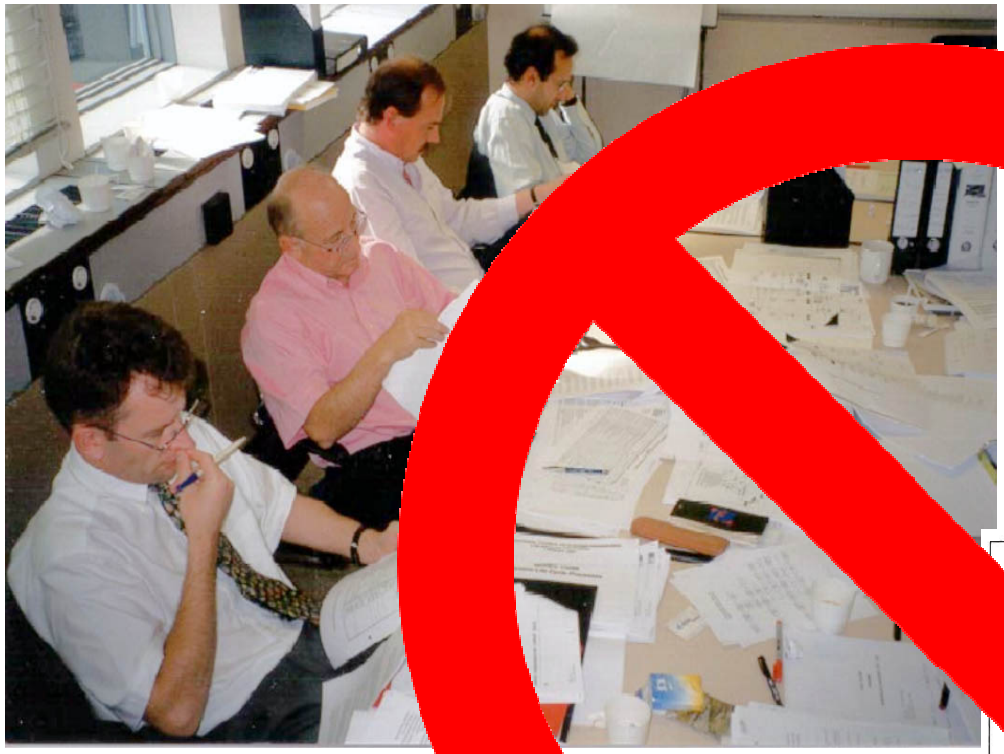

Let's build a smarter planet.

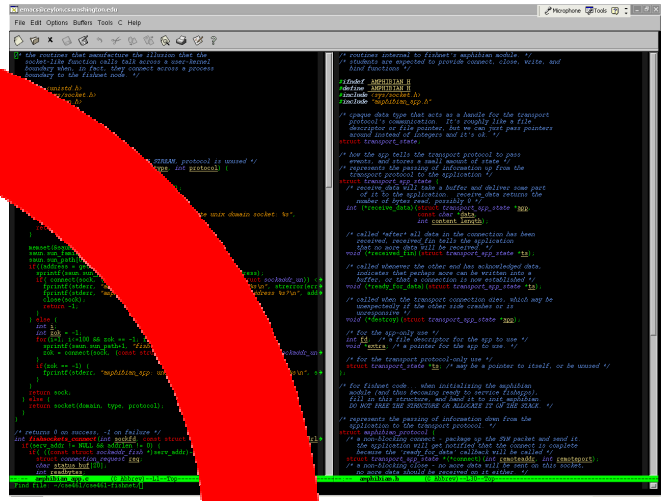
August 31, 2010 台北喜來登

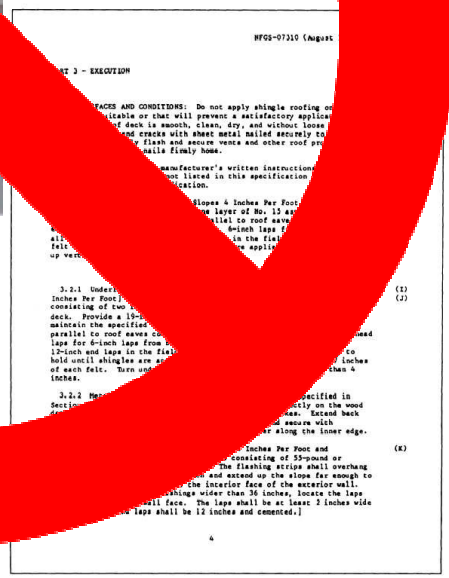
# The traditional design process



# Visual Modeling







하와이

The Korea Times Hawaii Edition

1998년 7월 17일 (우요일)

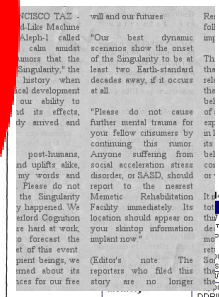
당뇨치료제 하와이 공급업자 모집

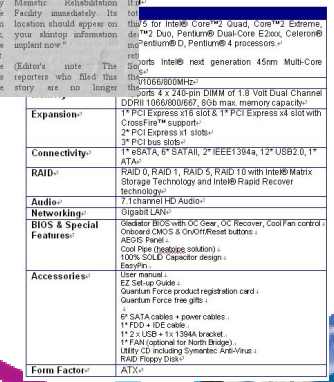
全国日本語学校データベースにようこそ!

このデータベースでは、全国にある400校以上のすべての日本語学校と日本語教育を行っている大学、短期大学、専門学校の情報を提供しています。データベースには、学校名や、コース、授業料、入学案内などの情報が含まれています。

なお、このデータベースに掲載されている日本語学校は、すべて(財)日本語教育振興協会の認定を受けています。

Watch phone

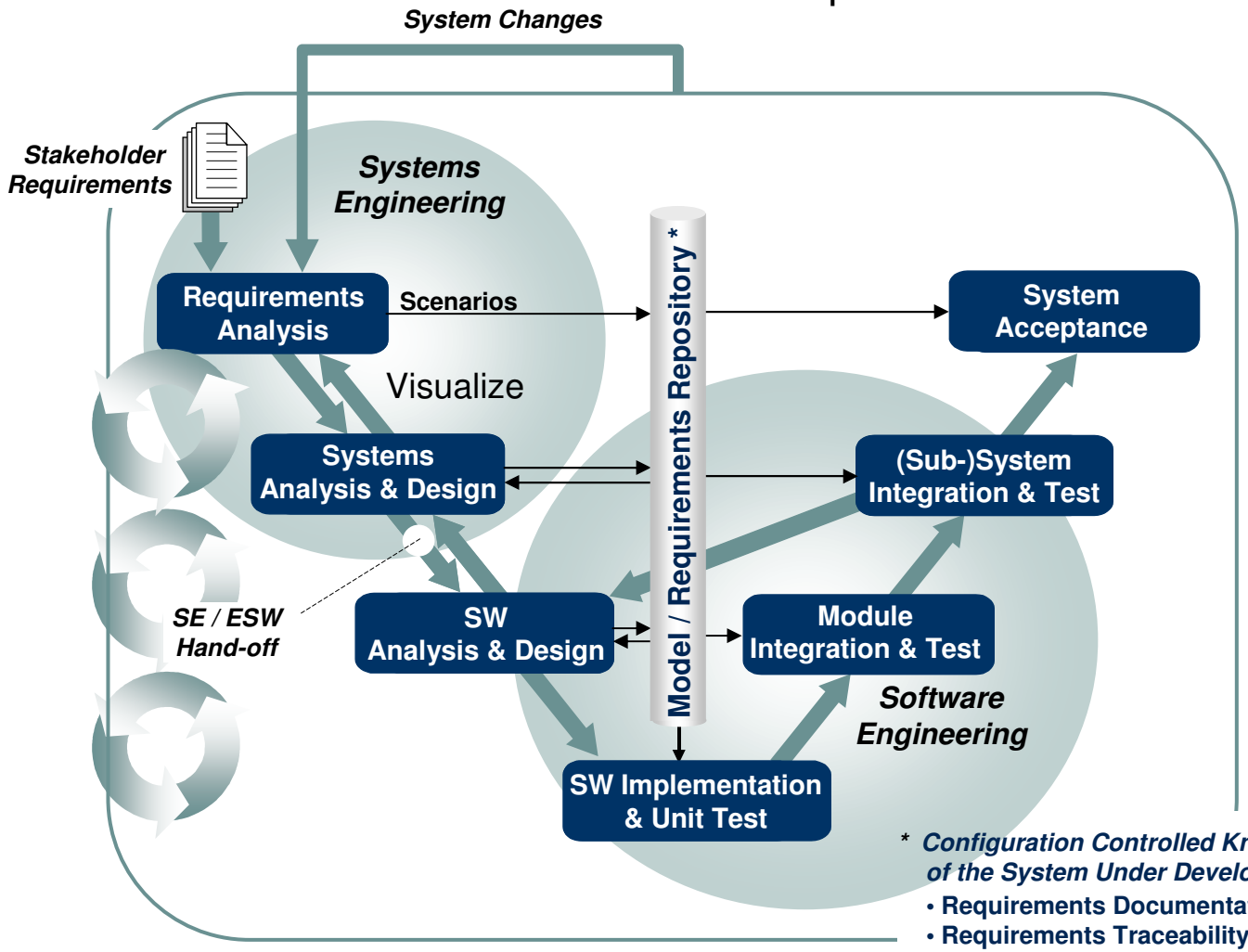




| | |
|---|---|
| Expansion | 1* PCI Express x16 slot & 1* PCI Express x4 slot with CrossFire™ support<br>2* PCI Express x1 slots<br>3* PCI bus slots |
| Connectivity | 1* eSATA, 6* SATAII, 2* IEEE1394a, 12* USB2.0, 1* ATA |
| RAID | RAID 0, RAID 1, RAID 5, RAID 10 with Intel® Matrix Storage Technology and Intel® Rapid Recover technology |
| Audio | 7.1channel HD Audio |
| Networking | Gigabit LAN |
| BIOS & Special Features | Gladiator BIOS with OC Gear, OC Recover, Cool Fan control, Onboard CMOS & On/Off/Reset buttons, ABOS Panel, Cool Pipe (heatpipe solution), 100% SOLID Capacitor design, EasyFlash |
| Accessories | User manual, EZ Set-up Guide, Quantum Force product registration card, Quantum Force free gifts, 6* SATA cables + power cables, 1* FDD + IDE cable, 1* 2 x USB + 1x 1394A bracket, 1* FAN (optional for North Bridge), Utility CD including Symantec Anti-Virus, RAID Floppy Disk |
| Form Factor | ATX |

# Integrated System / Software Development Process

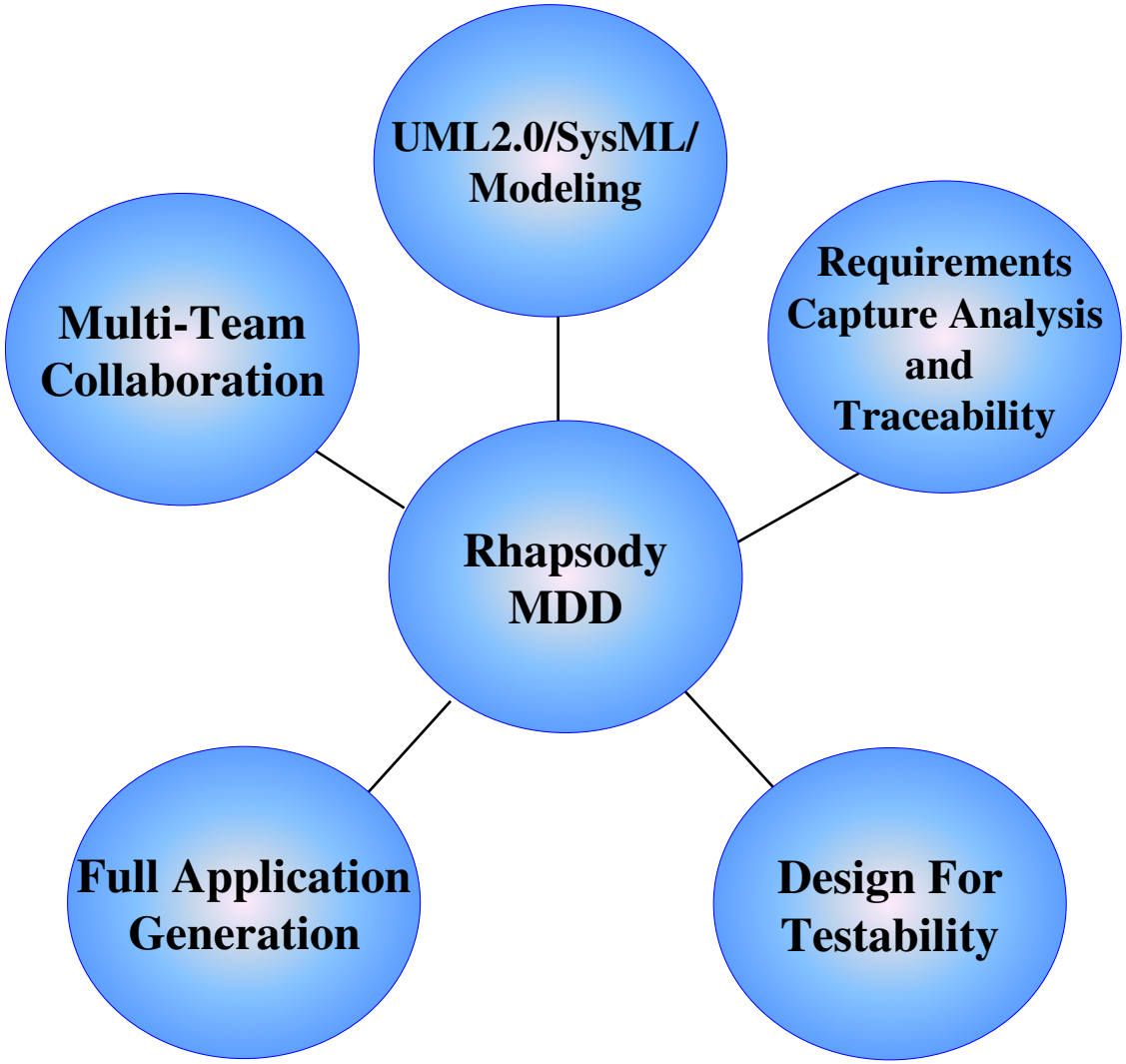
## Model-Driven Development




* *Configuration Controlled Knowledge of the System Under Development:*

- Requirements Documentation
- Requirements Traceability
- Design Documentation
- Test Definitions

# Marking All These Work




Rhapsody®

# Conceptual Collaboration in Text

## Developer 1

"Ok. Here's how it works. Thread A will p and that will change B's state to Running from v was Init. When B changes to Running it will sen and then wait for 2 second and then go back to Idle. Thread A will have started in Idle also and will go to Run after B sends back event Z which happens after the 2 seconds before going to Idle. All this should happen in less then 5 seconds."

## Developer 2:

"Huh ?" What are you talking about?

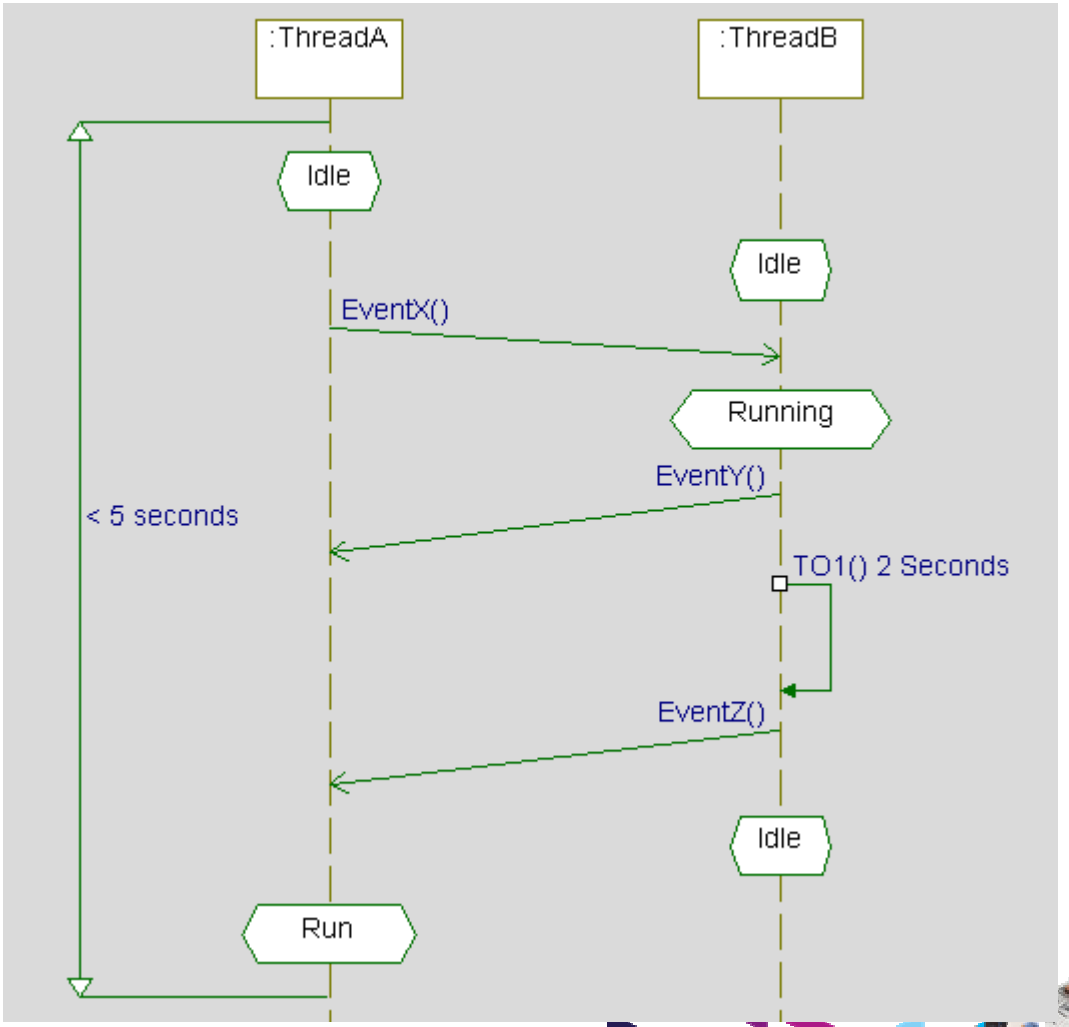
# Conceptual Collaboration in Models

Developer 1

"Here look at this Sequence Diagram."

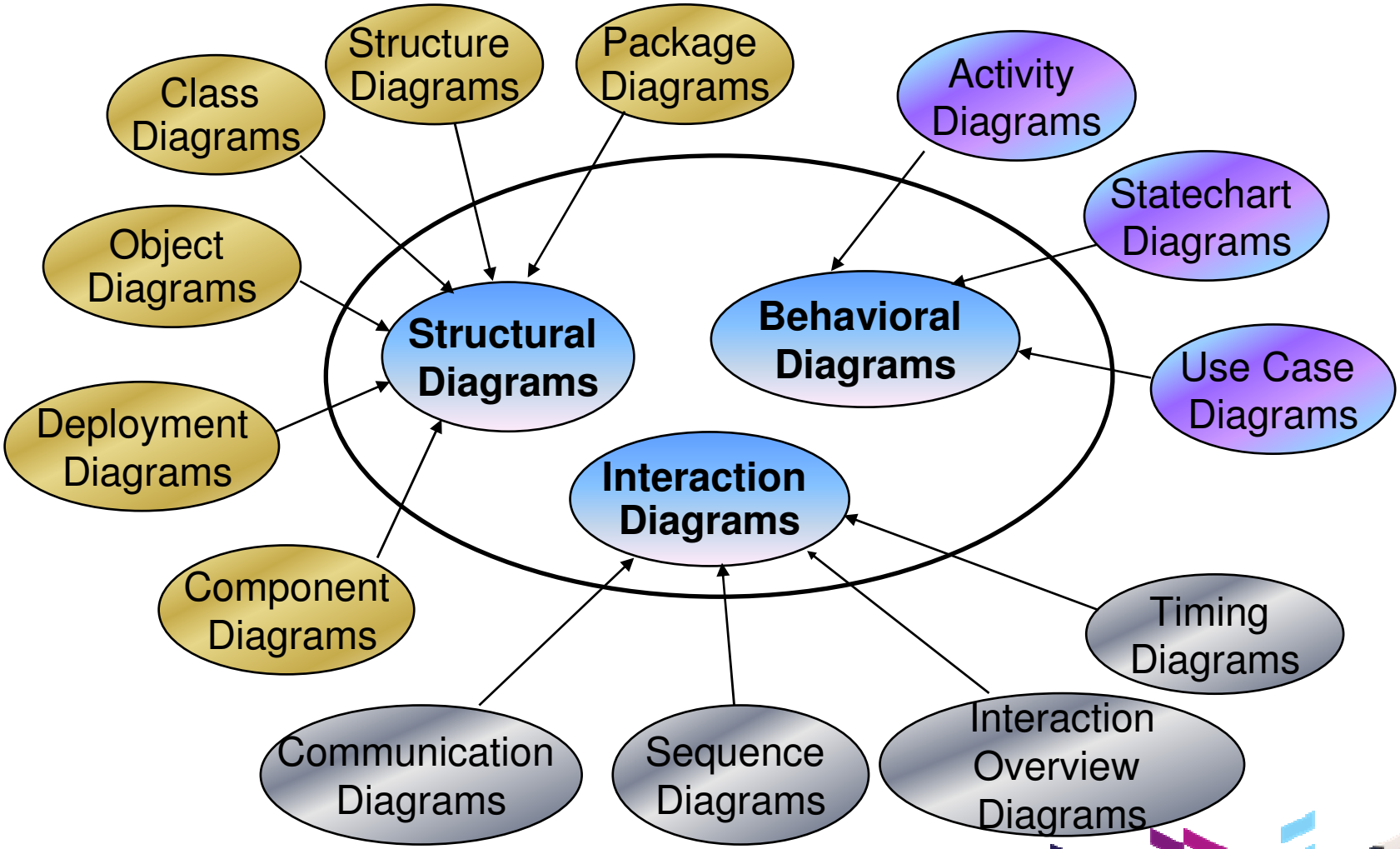
Developer 2

"Ahhh, now I see!"

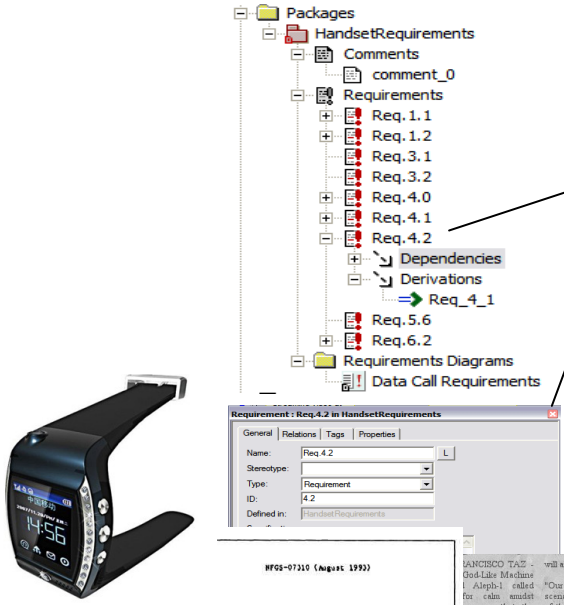



# Conceptual Collaboration in Models

UML2 - a common graphical language enabling conceptual collaboration



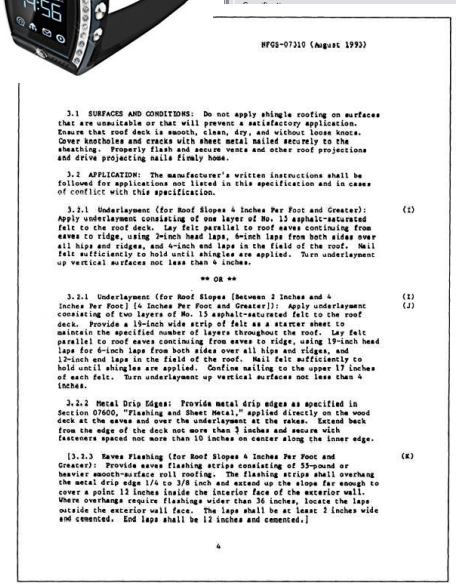
# Requirement Traceability

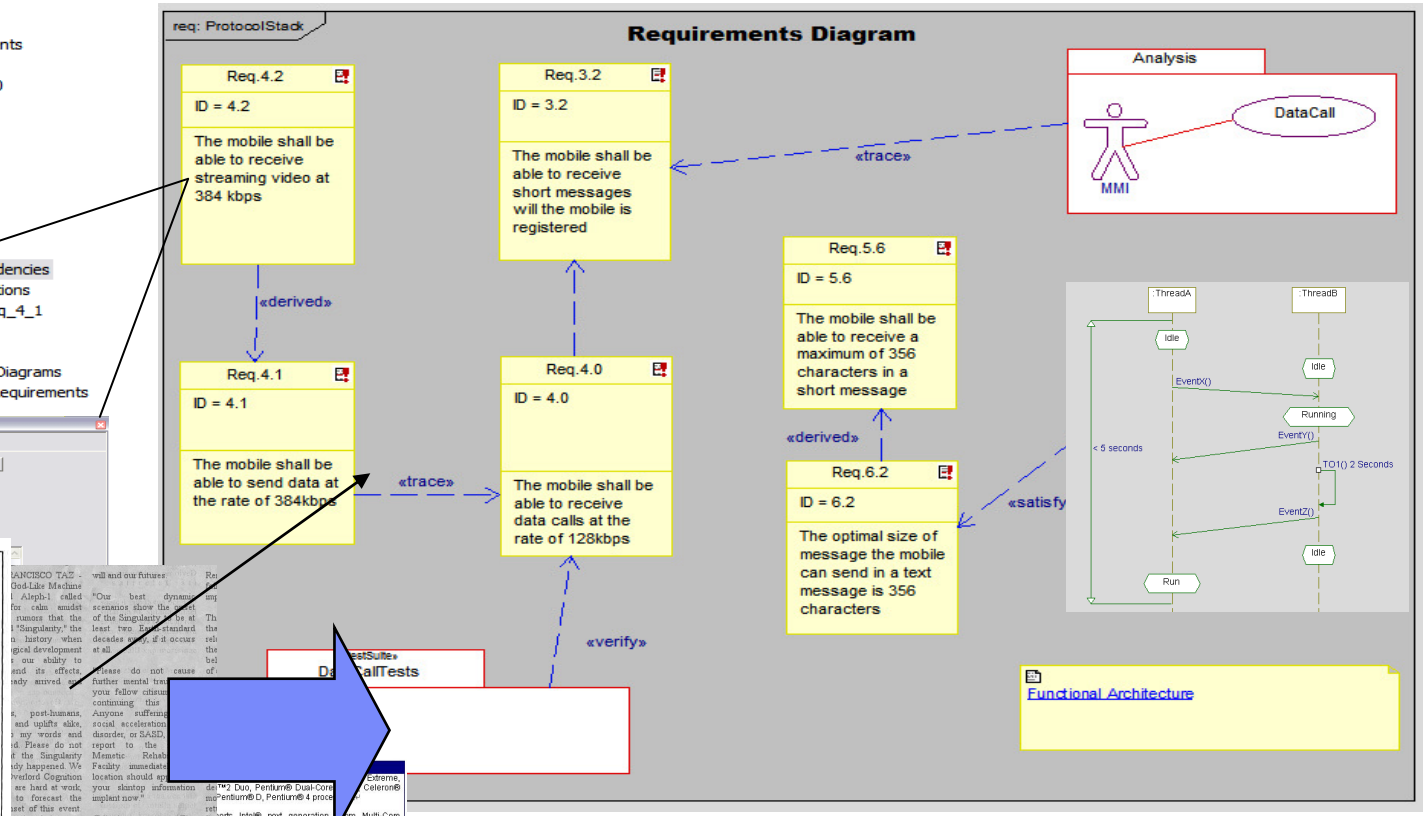


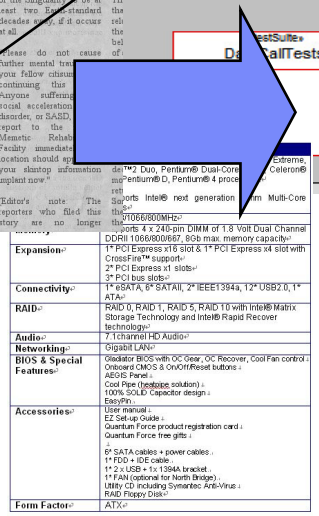




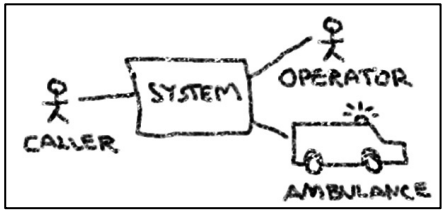




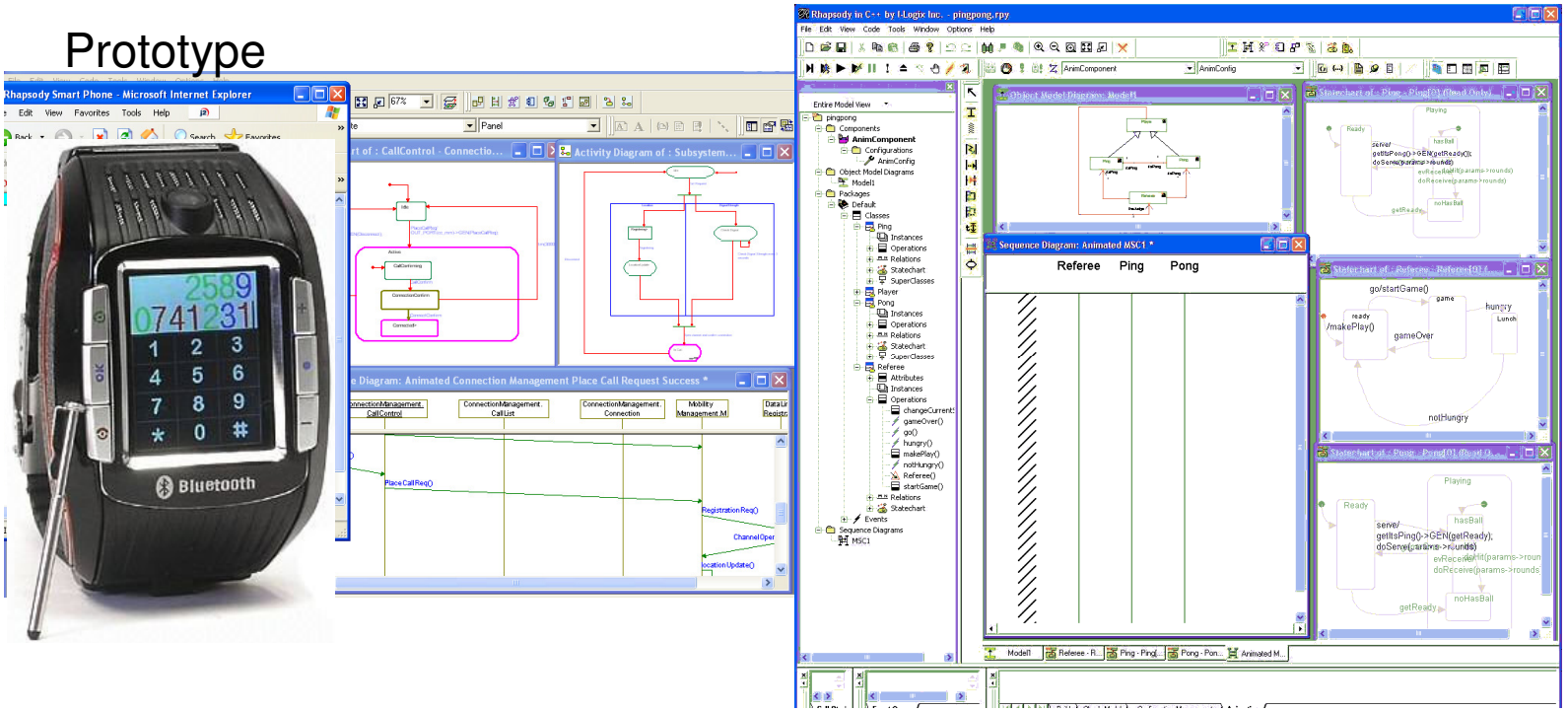

Coverage Analysis
Impact Analysis



# Executable models (Simulation and Animation)

***You can't test what you can't execute!***

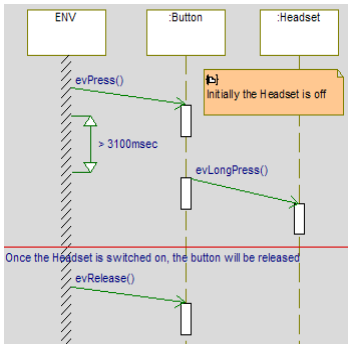
Prototype



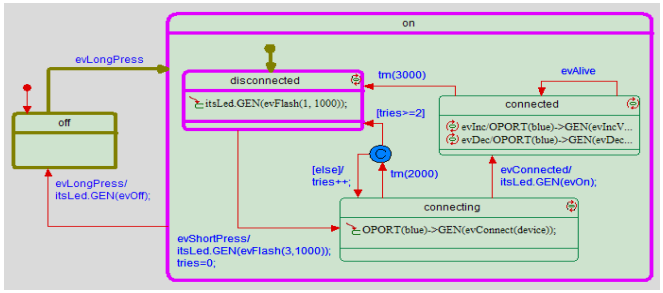
# Requirements Based Testing

- Use requirement scenarios to validation the design
- Automatically run multiple scenarios
- Easily identify errors

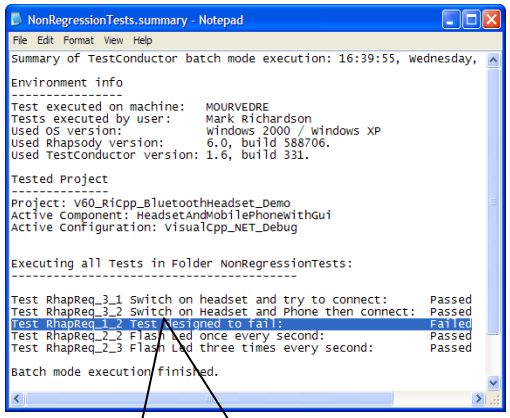
## Test Scenario




## Stimulate & Monitor the Model



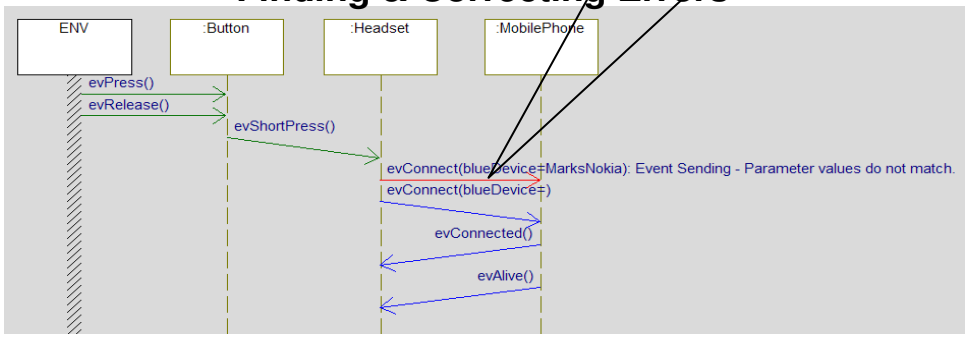

## Test Results



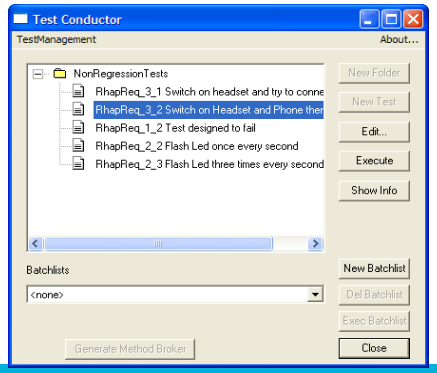

*Failed Test*

*Unexpected result*

## Finding & Correcting Errors




## Test Configuration



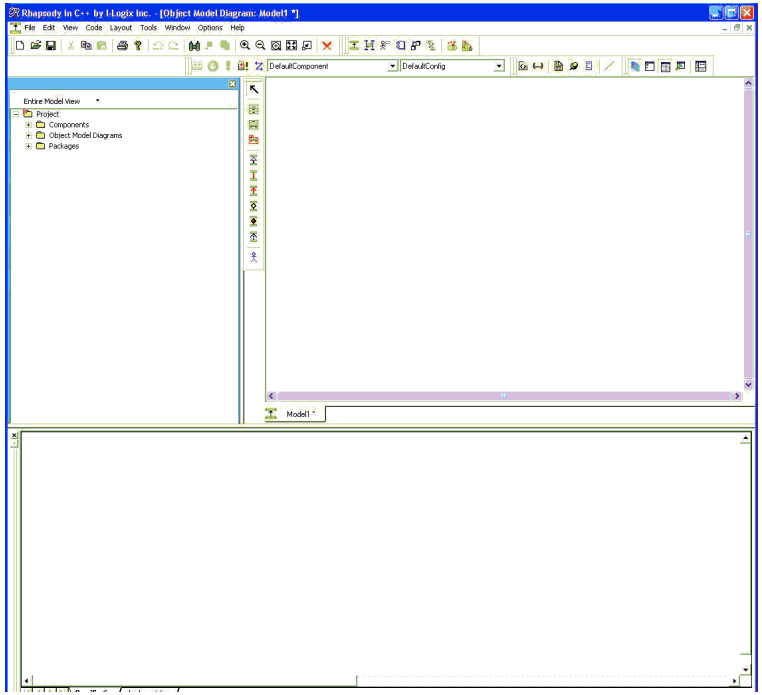
# Full Application Generation

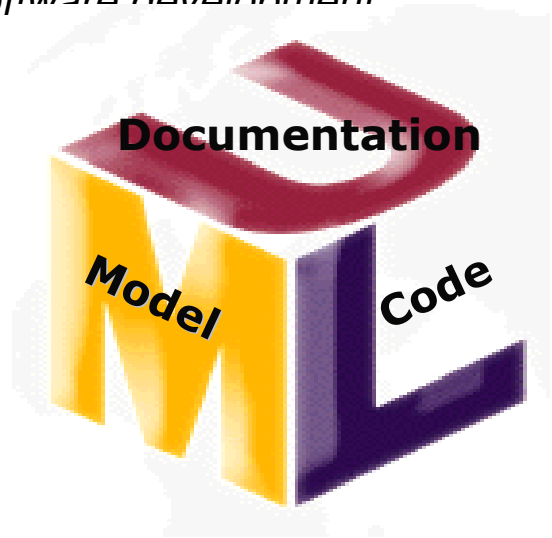
- Rhapsody leverages *all* structural and behavioral model views to produce an executable application
  - State machines: event driven behavior
  - Activity diagrams: algorithms and process flows
  - Generates all construction artifacts (e.g. Makefiles)
- Support for
  - C, C++, Ada and Java
  - Size/Speed tradeoffs
  - Coding style options
- **Seamless Reuse** of existing code and models (IP)
- **Dynamic Model Code Associativity (DMCA)** gives you the ability to work the way you want
- **The Real Time Framework** enables rapid application deployment onto any RTOS or systems with no RTOS

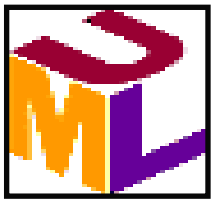
## Model Code Associativity

# *Rhapsody works the way you do*

- Design, Code and Documentation are always kept in sync
- Freedom to work at code level or design level
- Change one view, the others ***change automatically***
- *Critical for real-time embedded software development*

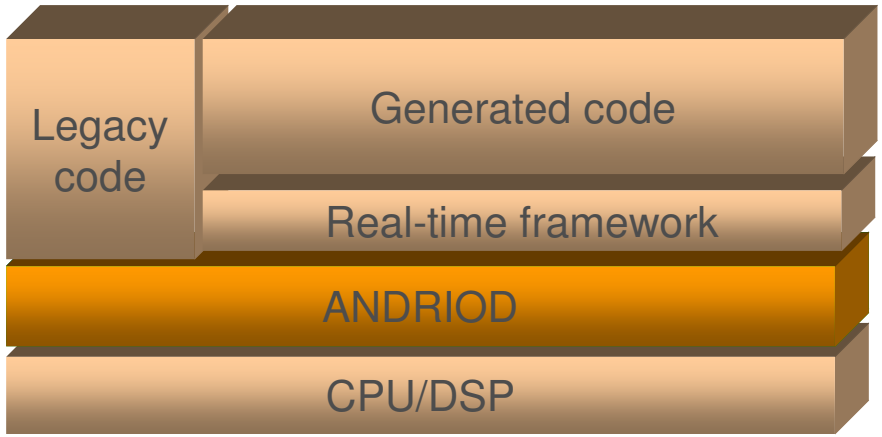




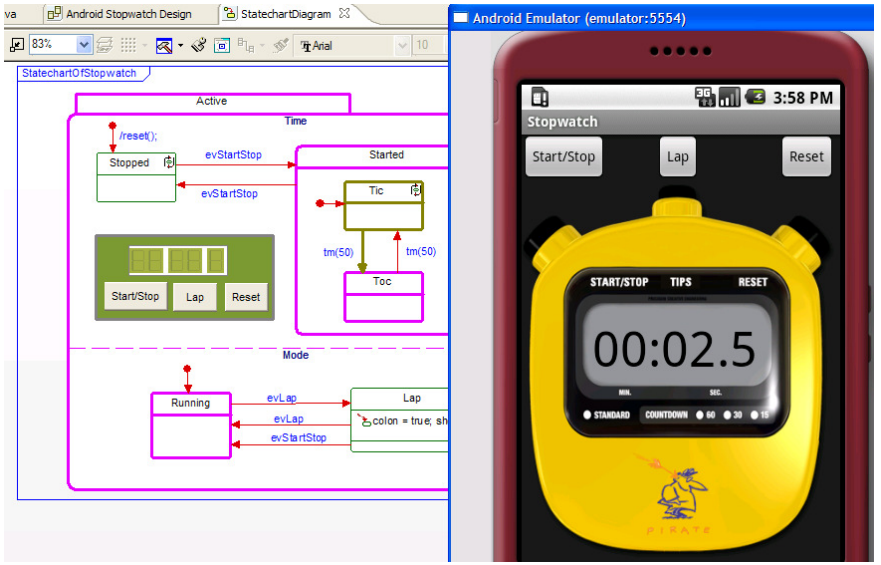
# *Rhapsody provides an executable real-time framework*

- Most applications are over 50% “housekeeping code” which is redeveloped every time you create a system.
- A *framework* is a partially completed application.
  - **you** customize and specialize for **your** application.
  - All source code is provided.
- A *real-time framework* is an:
  - integrated set of design patterns
  - optimized for embedded applications



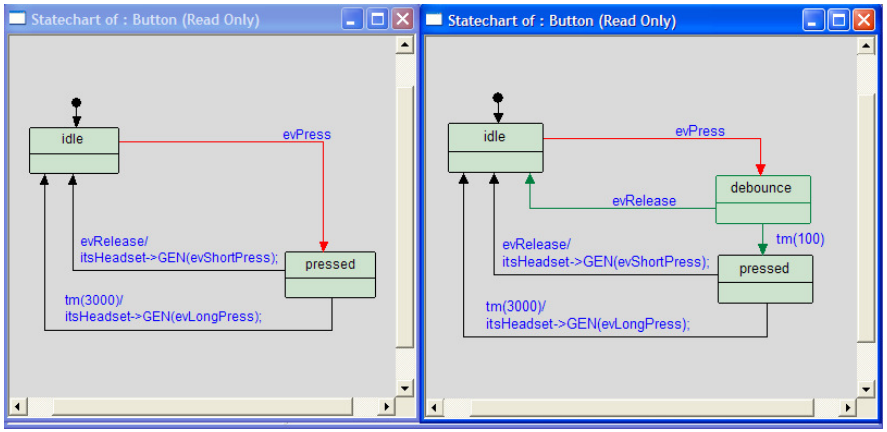
# Support for Android Development

- Android Model Library
  - Subset of the full fledged Android API
  - Help Rhapsody users to utilize Android's API in easy and visual ways
- Android Profile
- Automate integration with Eclipse/Android
  - Automate the steps in building the eclipse project
- Animation of Android Applications



# *Rational Team Concert enables distributed teams to perform as one through integrated collaboration, process and tools.*

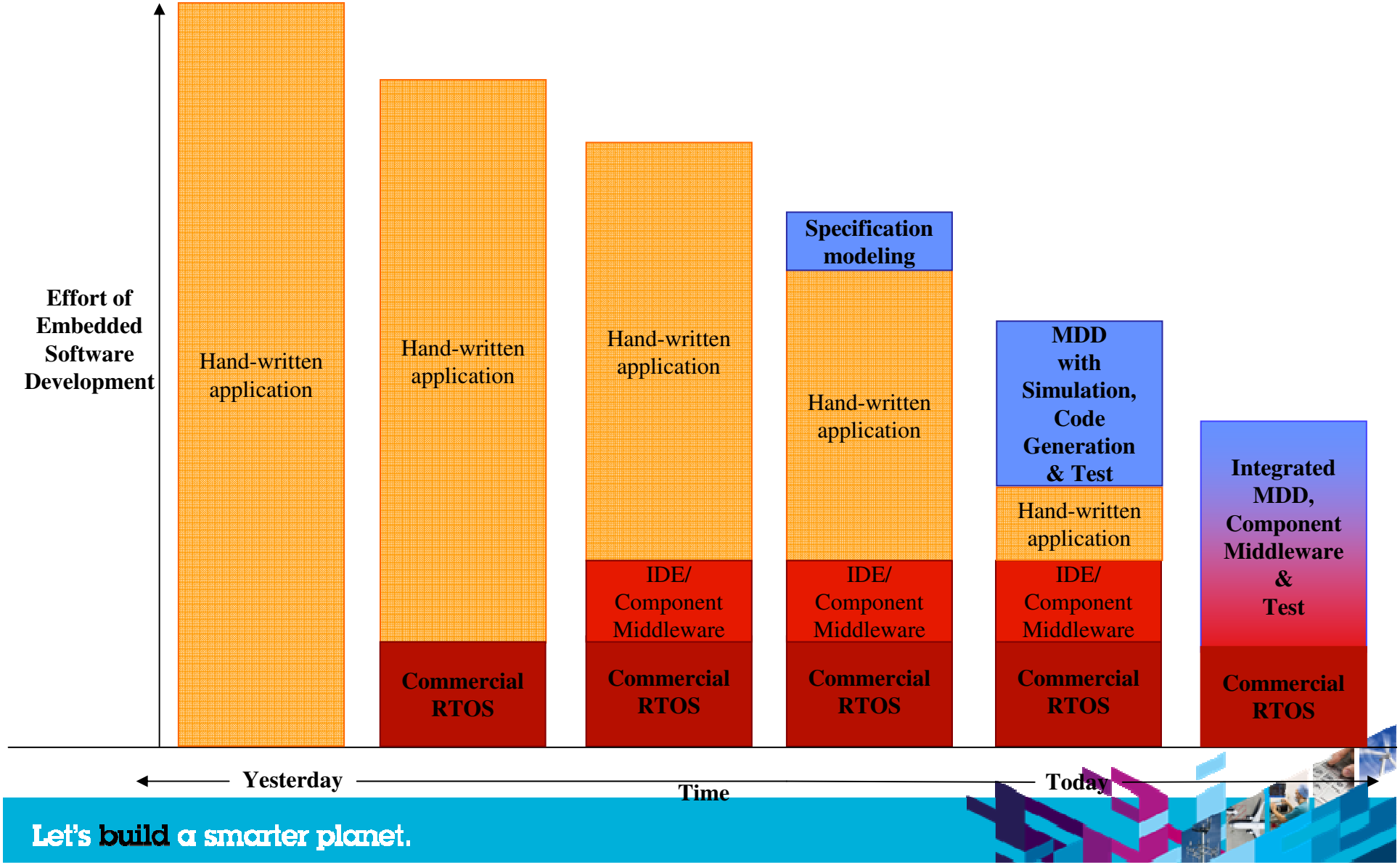
- Real time, in-context collaboration
  - Make software development more automated, transparent and predictive
- "Think and work in unison"
  - Integrated planning, source control, work item and build management




*Open and extensible on* Jazz

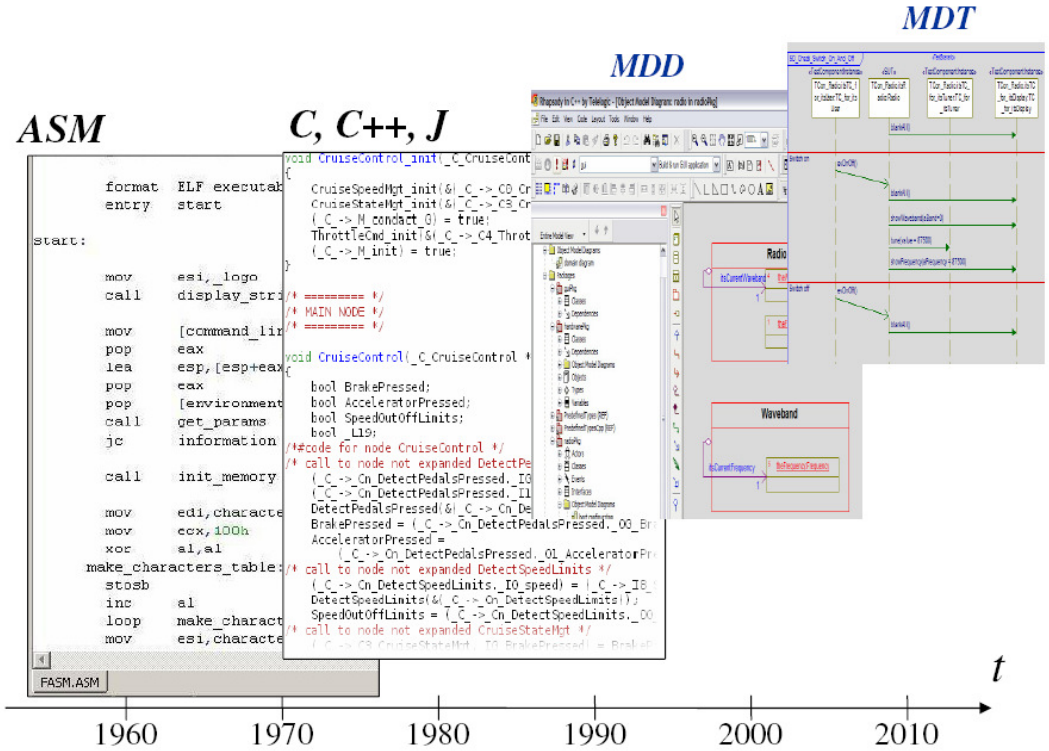
- ✓**Collaborate in context**
- ✓**Right-size governance**
- ✓**Day one productivity**

# Embedded Software Development Efficiency



# Embedded Software Development Efficiency

## Development Process is Evolving…



## Platform-Independent Models

- Intellectual property (IP) is expensive to create and maintain yet *crucial*
- Most embedded software must be recreated when moving to a new platform or environment
  - New Middleware
  - New source code languages
  - New Operating Systems
  - New Hardware
- Further, these systems must interface with massive legacy systems

## Platform-Independent Models

- The use of *PIMs* allows systems to be created that can easily be ported to new technologies, infrastructures and frameworks reusing corporate IP
- Intellectual property is then managed in a format more abstract and reusable than source code


Thank
You