

Model Driven Software Development, A Case Study The Good, the Bad and the Ugly

Gari Palmer

Senior Principal Software Engineer

Information Systems and Computing Technology Area

Director

May 18, 2011

MDSD Case Study: Agenda

- The Good, the Bad and the Ugly
- Model Driven Software Development – The Basics
- Case Study Scope and Method
- The Raytheon System
- The MDSD Components
- MDSD Summary Results
- MDSD Consultants
- Shortcomings of MDSD Program Use
- Advantages of MDSD Program Use
- New Technology Adoption Factors
- Towards a Culture of Change
- Suggested Improvements

The Good, The Bad, The Ugly

- *The Good*

- *Model Driven Software Development (MDSD) reduced development time, staffing and cost*

- *The Bad*

- *Model Driven Software Development was not fully embraced by the development and integration teams*

- *The Ugly*

- *Model Driven Software Development has not gained the adoption we would like to see*

There are many sides to the MDSD story

Model Driven Software Development – The Basics (1 of 3)

Traditional
Software
Development



Model Driven Software
Development

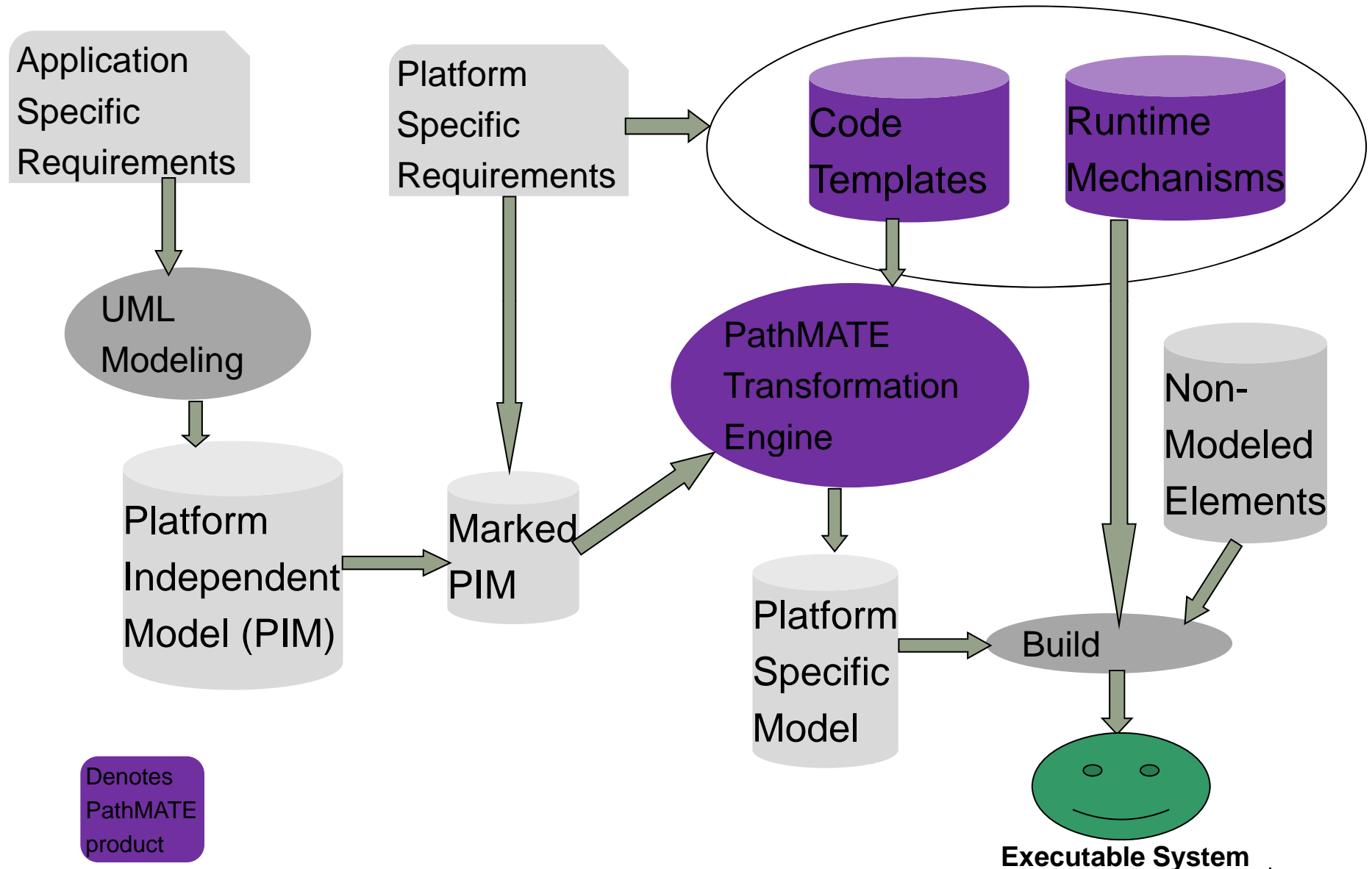


Model Driven Software Development – The Basics (2 of 3)

- Model-Driven Software Development is the term used for defining systems, including behavior, in models, and then using the models to generate deliverable code
- Platform Independent Model (PIM) of an application's functionality and behavior
- Developers mark up the PIM with platform specific notations
- Models transformed to code using standardized mappings for specific target platforms (can be provided by mature tool such as PathMATE by Pathfinder Solutions)
- Models transformed to Software Design Document
- Design and code are always synchronized

MDSD raises the level of abstraction

Model Driven Software Development – The Basics (3 of 3)



Case Study Scope and Method

■ Why:

- Uncover the pros and cons of MDSD use on a program
- Provide insight to how we can improve MDSD deployment
- Communicate to engineers and managers

■ What:

- A retrospective of the deployment of MDSD on one Raytheon program
- Interviews with 12 people:
 - Architects
 - Software Developers
 - Integrators
 - Program and Software management
- Productivity and defect density metrics were collected from a Six Sigma project

Information in this presentation is from the interviewees and Six Sigma Report

The Raytheon System

■ System Description:

- A weapon system used against:
 - cruise missiles,
 - unmanned aerial vehicles (UAVs)
 - fixed –wing and rotary-wing aircraft
- The system integrates surveillance, command and control, fire-direction, fire distribution and engagement capabilities
- The system is currently delivered



The MDSD Components

- The Communications Architecture. Two of the message handlers (in green) were generated with MDSD
- The existing interfaces were reused from another contract



MDSD Summary Results

- Positives:
 - Generally considered a success by managers and engineers
 - The customer was extremely happy with working components developed under budget, on time with lower defect densities
 - Under ran budget
- Negatives:
 - Abstract development approach reduces understanding of system details
 - Harder to find the origin of a defect during integration
 - Extensive involvement from Pathfinder Solutions consultants

The Customer: “The Software Organization beat the budget
– a refreshing change”

MDSD Summary Results Metrics

- Calculation approach:
 - Raytheon measures for productivity and defect density are based on Source Lines of Code (SLOC)
 - MDSD code generation typically results in greater SLOC than traditional hand coding
 - MDSD generated SLOC count was decreased by 50% to normalize measures which reduces productivity and defect density results
 - MDSD Consultant costs and Developer training are embedded in Design, Code and Unit Test program costs
- Measures:
 - Planned for traditional coding approach. Only used **65%** of planned **developers**
 - Design, code, unit test and integration (DCTI) **productivity** for both Message Handlers was at least **44% greater** than standard
 - **Defect Density** (defects per KSLOC) was **1/3 business average**

MDSD Consultants

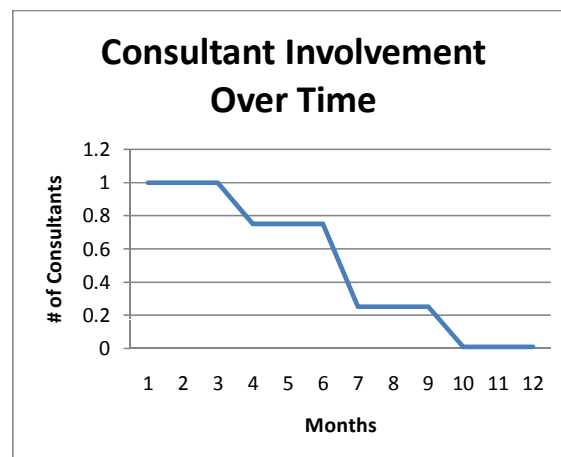
- Pathfinder Solutions (vendor of PathMATE) highly recommends the use of consultants to get a project off on the correct path

Pros:

- High caliber consultants
- Available for quick fixes to PathMATE (pro & con)
- Provided training and mentoring on architecture methodology, OO design and PathMATE specifics

Cons:

- If the tool were more stable, there was better documentation or Raytheon had more expertise, there would be less need for consultants
- Use of consultants side by side with developers calls productivity into question
- The use of consultants may not be scalable for deployment to Raytheon



Shortcomings of MDSD Program Use

■ Tool:

- Debugging during integration is more complex
- Need fairly extensive experience with the toolset to make it work properly
- Training and mentoring was required for success when developers preferred to write code
- Very simple changes may require knowledge of multiple tools rather than just a programming language and compiler
- VxWorks integration was immature

■ People:

- Not all engineers adapt well to new methods and levels of abstraction
- Hands-on training is best for comfort with new tools
- Mentors must be available
- MDSD is built on OO. Therefore, a solid OO foundation is beneficial
- Lots of communication is required

Advantages of MDSD Program Use

- Trivial task to make some global changes
 - Changed 72-word message format to and 80-word message format for hundreds of messages in **1.5 weeks**
- Application code generated by MDSD resulted in lower defect density
- Design and code are always in sync
- Method enforced common vocabulary and design guidelines
- Collaborative approach to architecture and design

Program Office Quote “The customer was so impressed that for a year the MDSD success was mentioned in their viewgraphs”

New Technology Adoption Factors

- Risks and opportunities must be communicated
- The advantages and disadvantages must be communicated
- Expect challenges with technology when it is new to the development team
- Not every engineer is ready for the challenge
- Hands on training for all team members is key
- Mentors/consultants need to be available
- Must communicate successes (productivity and quality) to team

The Defense industry and it's partners are risk averse – we still need to make forward progress with new technologies

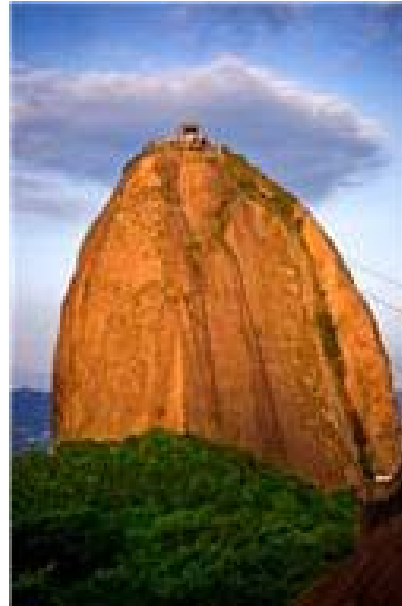
Towards a Culture of Change

- At a Corporate level, Raytheon realizes that new technologies drive better productivity and products
- Raytheon has been adopting more technology-based development paradigms (Agile, Lean, MDSD, Domain Specific Languages, Software Factories)
- Raytheon software management respects and rewards technology adoption

Suggested Improvements for Technology Deployment

- Select teams based on their experience and willingness to engage with new technology
- Ensure effective training for all engineers no matter when they join the project
- Existing processes and measures do not always neatly map to new technologies. New processes and measures need to be incorporated to foster wider adoption
- Continue to investigate new MDSD tools and technologies
- Encourage engineers and managers to understand MDSD benefits and pitfalls
- Provide feedback on productivity, schedule and quality status to developers

Summary



Methods

Tools

Training

Communication

SUCCESS

Acronyms

- *MDSD – Model Driven Software Development*
- *OO – Object Oriented*
- *SLOC – Source Lines of Code*
- *UML – Unified Modeling Language*