What Do You Do When the Horse You're Riding Drops Dead?

Why Model Driven Design is Emerging as a Preferred Best Practice

By Jerry Krasner, Ph.D., MBA

March 2007

EMBEDDED MARKET FORECASTERS

American Technology International, Inc.

Background

Embedded designs have become increasingly more complex while the windows of opportunity continue to shrink. In the global world of embedded enterprise, it is not unusual to find OEMs and systems integrator design teams spread out by geography, operating in different time zones, experiencing overlapping areas of responsibility and finding bugs much too late in the design cycle.

The resulting design delays are expensive (2006 EMF data shows a consistent 4-month average delay with 42% of all embedded designs behind schedule) 10.4% of embedded designs are cancelled – and the average time-to-cancellation is 6-months. Engineering time associated with such delays and cancellations is very expensive – and does not include missed opportunity costs.

Annual surveys by Embedded Market Forecasters (EMF) of embedded developers have shown that software development is responsible for more than 80% of design delays and associated design complications. This data also reports on embedded developer responses to design complications. When asked how close their final design was to pre-design expectations (for performance, systems functionality, features and schedule) approximately 40% of respondents indicated that their final design was NOT within 20% of their pre-design expectation.

Whether the system is poorly conceived, specified or whether crucial algorithms fail to adequately address systems performance, traditional methods of embedded software development are yielding to a process known as “Model-Driven Development™ (MDD™)”. MDD is used to more clearly define design specifications, test systems concepts and to automatically generate code for rapid prototyping as well as for software development.

One of the major advances in software engineering design has been the use of the Unified Modeling Language™ (UML®) for enabling embedded design efficiencies. Pioneered by companies IBM/Rational, Telelogic and the former I-Logix for embedded and/or real-time applications, its value is the ability to address very complex designs and (with certain commercial offerings) the ability to go from State Diagrams to source code (automatic code generation). In addition, UML offers code reuse, legacy upgrades and the ability to re-port software to changing hardware configurations. This is of huge importance given the frequent end-of-life changes in chip availability.

Introduction

Embedded designs are becoming more complex and marketing-based windows of opportunity place a premium on the efficiency and effectiveness of the design process and of design outcomes. EMF studies regarding purchasing decision-making illustrate a further problem. The acquisition process of operating systems and development tools is most frequently detached from the financial consequences of being late to market (and the increased engineering and lost opportunity costs) and are usually based on tools familiarity and up front acquisition costs.

Such considerations as developer productivity, product upgrade capabilities and component reuse have in the past been hard to quantify and are seldom considered in the acquisition process.

There is a very clever children's story regarding what one can do if the horse your riding drops dead – and the absurd options have been used to illustrate similar decision making efforts that have been set forth as "solutions" to business strategies, administrative problems and herein for what technical managers do when their product design has fallen apart and critical decisions need to be made.

Along with this theme, we will present several solutions set forth for dealing with the dead horse. We leave it to the reader to fill in the comparable solutions that their engineering teams use to deal with crucial design situations.

Clearly, the moral of the story is the need for pre-design planning and flexibility.

Your horse has died while you were riding it – what do you do?

- Appoint a committee and hire consultants to study the dead horse
- Arrange to visit other sites to observe their *best practices* in riding a dead horse
- Form quality and performance teams to make dead horses run better
- Harness several dead horses together for better efficiencies
- Hire a more experienced rider and give him a stronger whip
- Establish a quality circle to find users for dead horses

Flippancy aside, the steps taken have a strong resemblance to the post-mortem design evaluation committees that most of you have experienced.

The best design practices involve the following:

- Select design tool sets and methodologies that can be shown to increase developer productivity

- Base the selection process on data that suggests early design completions and minimization of late completions.
- Ensure that final design results closely coincide with pre-design expectations
- Select solutions that offer effective design reuse
- Select solutions that offer extreme flexibility to reconfigure design objectives without totally disrupting the design effort.
- Select a solution set that makes it possible to move developers between different projects without having to learn how to use a new set of tools

Currently, MDD is the only methodology that meets these objectives, and the data will show that developers that use UML get the most productivity and best design outcomes.

Looking at the Data – Results of EMF Surveys of Embedded Developers

Productivity

The term productivity is widely used and widely misinterpreted. Questions arise as to how it is measured and what the measurements mean.

As a matter of comparison consider that 100 years ago Americans spent 43% of their income on food and another 14 percent on clothing. At that time 41% of Americans worked on farms. In 2002 the figures for food and clothing dropped to 13% and 4% respectively while family incomes (adjusted for inflation) tripled. Because of advanced technology, management skills, new fertilizers, etc., 2% of the US population has been able to feed not only all Americans, but the rest of the world too.

This productivity has enabled our discretionary incomes to support our purchases of HDTV's, boats and fancy cars, etc. So how do we measure developer productivity and what measurable benefits does it offer?

About the survey

Embedded development engineers were interviewed via a comprehensive survey designed to elicit information regarding current and anticipated tool usage, design starts, completions and cancellations, development (host) and target platforms, microprocessors used, desirable and undesirable product features, vendor evaluation criteria and purchasing decision processes, among other important information. Responses from 510 embedded developers comprised the EMF 2006 comprehensive survey, which explored the attitudes, preferences and values of embedded developers to the current and projected use of embedded technologies. The survey was constructed such that the responses could be evaluated from many perspectives, including total response, specific job title of the respondent, architecture employed in embedded design, processor family, and each of ten embedded vertical market application (e.g., telecom, industrial controls, etc.).

The survey questionnaire was developed jointly by Embedded Market Forecasters (EMF) and Wilson Research Group and programmed for a web-based survey deployment by Wilson Research Group. A base of nth-name selected subscribers to RTC magazine, COTS Journal, and Military and Aerospace Electronics, who indicated on a qualification card a professional area of interest in embedded development, were used as the sample. Starting on 6 March 15, 2006, each member of this sample was sent an email explaining the purpose of the study, that they were statistically selected, and that their participation would help embedded vendors better service the developers. The database was identifiable by a PIN number that respondents used when they went on-line. This served to insure that the appropriate individual responded to the survey and that the response was one-time only. Care was taken to insure that there was no duplication between the two groups. One follow-up email was sent to all non-responding members of the sample two weeks after the original mailing.

Five hundred and ten developers responded to the online survey, of which 55 were hardware engineers, 130 were software engineers, 97 were systems engineers, 52 were systems architects, 43 were firmware engineers, 36 were software engineering or development managers and 35 were hardware engineering managers. In addition 62 developers gave titles other than these listed. This provided an excellent distribution of experiences and viewpoints from which to draw inferences and conclusions. Statistically, the response is at a 95% confidence level, plus or minus 3.6%.

The EMF survey of Embedded Developers asks developers to provide the number of "new to world" design starts and "upgrades to existing products" design starts that they have participated in. Table 1 shows the growth of UML-based design starts as compared to the industry average.

	2004		2005		2006	
	Industry	UML	Industry	UML	Industry	UML
New to world	1.5	2.1	2.6	4.6	3.1	6.8
Upgrades of existing designs	2.2	2.8	2.8	3.8	3.6	9.4

Table 1:
Comparison of Industry Average Design Starts to UML Design Starts

From Table 1, we see that developers that use UML in their designs are able to manage (year over year) more design starts and completions than the industry average. This translates into higher productivity per developer and greater savings to the organization.

Another measure of productivity is to look at the number of lines of code written per developer – understanding that UML developers have automatic code generation available to them.

The EMF survey asked developers to indicate:

1) Total lines of hand-written code
2) Total lines of code per project

Let's look at the efficiency and productivity of developers by examining the average written lines of code per developer per project between the industry average and UML users as presented in Table 2.

	Industry Ave	UML user
Lines of hand- written code per project per developer	7514	4315
Total Lines of Code per project	667600	710000

Table 2:
Comparative Efficiencies in Project Lines of Code Written

Here we see that UML developers are required to hand-write 42.6% less code per project than the industry average. Table 2 also shows that this holds true for comparable size project. Because of the leverage of automatic code generation,

UML developers are required to hand-write less code (and hence experience fewer bugs)

Comparison of Final Design Results to Pre-Design Expectations

There are other measurements that are of importance. No matter how productive UML users might appear to be, the quality of their design outcomes must be at least as good as the industry average.

Table 3 presents three critically important measurements that EMF uses to determine design outcome. They are:

1) Percent of designs that are behind schedule
2) For behind schedule designs, the average number of months behind
3) The number of months from design start to shipment

	Behind Schedule	Months Behind	Months Start To Shipment	Total (ave.) Lines of Code x1000
MDD developers	35.1%	3.7	15.8	713.6
Non-modeling developers	44.3%	4.2	13.6	649.4
VxWorks users	49.0%	4.3	16.9	699.1
Lines of code <100k	43.3%	3.5	12.4	
Lines of code > 100k	44.1%	4.8	16.5	
Lines of code > 500k	46.2%	5.5	17.5	

Table 3:
MDD versus Non-MDD Design Outcome Comparisons

In Table 3 we present the average of the three leading MDD tools and compare them with non-modeling tool users and the market leading Wind River Systems tools. The industry average is broken out between three levels of code length. Obviously, we would expect better design outcomes for projects that have the fewest number of lines of code.

Extensive EMF surveys show that the average number of software developers per project is 25.7. If the average cost per engineer (salary, benefits and supporting technical staffing) is only $150,000 per year, then the average cost per each month of delay is $321,250.

Following this analysis and knowing that EMF data shows that on average 42% of embedded designs are behind schedule – an average of 4.1 months – we can calculate the average expected loss due to design delays per project as ($321,250)*.42*4.1 = $553.200. This of course does not take into account the cost of lost opportunities nor the impact on corporate overhead.

The bigger point made here is that spending slightly more on a better tool suite repays users over and over because they can deliver their designs on time; conversely, the competition completes their design late because they "saved money" by selecting an inferior development tool. EMF strongly recommends that tool selection analysis be made a "best practice" for embedded development projects.

Furthermore, any design outcome analysis must consider the closeness of the final design results to the pre-design expectation. EMF estimates that any final design not within 30% of its pre-design expectation be considered a poor design outcome.

Figure 1 illustrates the productivity (design accuracy is considered a productivity measure) outcomes of MDD and non-MDD designs.

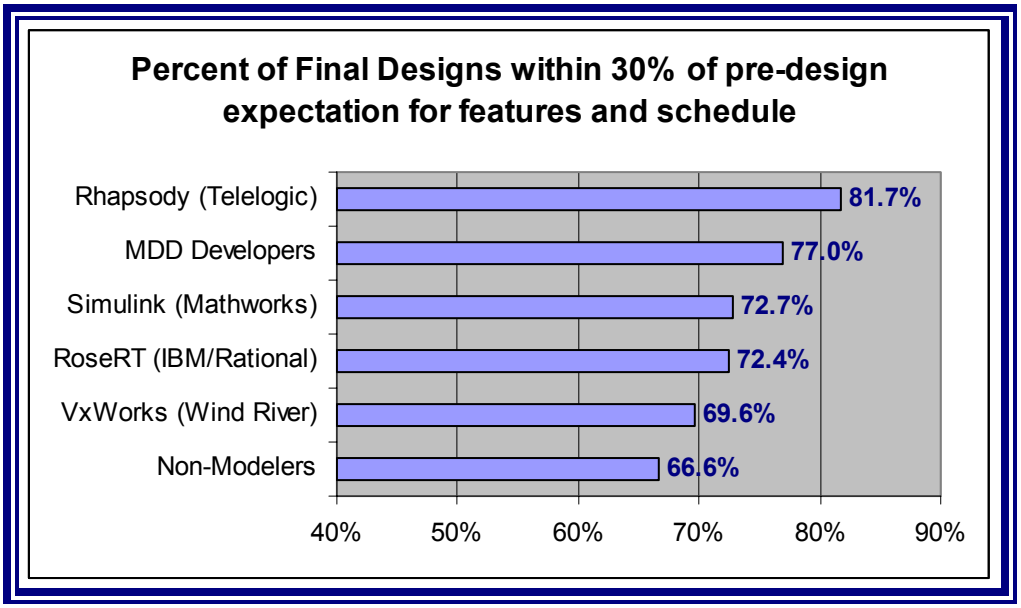

Figure 1

Clearly MDD developers enjoy a substantial outcome result over non-modeling developers.

Leaders & Laggards

EMF working together with Lone Star Aerospace and their CEO Steve Roemerman undertook a different design outcome analysis – one that was able to look at the design practice differences between developers that exhibited sustained best outcome results (Leaders) and those that exhibited the worst sustained outcome results (Laggards).

Lone Star Aerospace (LSA) is a highly regarded organization serving defense and aerospace markets with strategy, systems, and electronics offerings. They are a principle consultant to NASA, the US Navy, Army and Marines and to other the larger prime contractors. LSA conducts surveys of influential military and government officials regarding technology acquisition and deployment. LSA has used the EMF database as the exclusive source of design activities and outcomes in its consulting endeavors.

An analysis of the differences is presented.

Leaders are more likely to use modeling/simulation and spiral development; Leaders are 30% more likely to use modeling and simulation than Laggards. Leaders have about a 7X better chance of making or beating schedules. Leaders are about twice as likely to use Automatic Code generation.

1. Leaders have slightly shorter projects
2. Leaders have a population with a ratio biased more to software developers
3. Leaders have teams that are about 25% Smaller
4. There are about 2.5 times more Laggard organizations than Leaders, and Laggard organizations appear to make up nearly half of the suppliers in most verticals, including DoD.
5. Leaders have about a 7X better chance of making or beating schedules
6. Leaders have about one forth as many late projects per work location
7. Laggards spend about 8 times more money addressing schedule problems
8. Leaders are more likely to develop products, or update products to include new features or take advantage of new processor features.
9. Laggards are more likely to develop or update products to address problems experienced in prior generations
10. Outsourcing:
    - Leaders are more likely to abstain from software outsourcing
    - Laggards are twice as likely to outsource the majority of their development
11. When they are late, Leaders are only about 60% as delayed as Laggards
12. When a development is troubled:
    - Leaders are more likely to make minor mods to the hardware or software, than Laggards
    - Leaders are almost twice as likely to change development toolsets

- Laggards are more likely to do a major software re-write
- Laggards are more likely to remove product features
- Laggards are more likely to slip schedule
- Laggards spend about 30% less on personal tools per developer
- Laggards spend about 30% less on a site wide basis for tools

13. By a number of measures, tool spending is the only "economy" seen with Laggards – laggards spend about 30% less
14. Leaders are more likely to use modeling/simulation and spiral development; Leaders are 30% more likely to use modeling and simulation than Laggards
15. Laggards are more likely to use code walk throughs
16. Leaders use UML 2
17. Leaders are more than twice as likely to use device driver tools
18. Leaders are about twice as likely to use Automatic Code generation
19. An average Laggard organization spends over $40 million a year more, in order to achieve results that are never superior by any measure we tested. Based on the size of the Laggard business unit population, this suggests that Laggards waste several billion dollars a year of tax payer money

Summary

EMF data – derived from extensive interviews and survey results of embedded developers shows that:

- MDD is the fastest growing segment of the embedded marketplace year-over-year since 2004
- MDD produces better design outcomes (time to market, on-time delivery, proximity of final design results compared with pre-design expectations)
- The Joint Chiefs of Staff have a UML modeling system in place as does many military and government groups. Systems-within-systems modeling has become a major focus of military forward thinking and MDD is seen as key to achieving success
- The major movement within government funding is emerging as “systems within systems” design and analysis. MDD is a key component
- We have interviewed the engineers at a major prime contractor separate from the usual EMF survey respondents – but using the same survey Programs that used MDD had better design outcomes than those that did not use MDD

EMF data has also shown that:

- The use of simulation-modeling tools by embedded developers has reduced design delays and cancellations
- The use of simulation-modeling tools by embedded developers has significantly improved the relationship between pre-design expectations and final designs
- UML is the most popular graphical representation for simulation-modeling tools for discrete embedded system designs
- UML enables faster design iterations that produce desired performance, functionality and capabilities
- Using UML, design cycles are more predictable and result in faster product shipments
- UML contributes significantly to a reduction in design, development and implementation costs

About the Author

Jerry Krasner, Ph.D., MBA is Vice President and Chief Analyst of Embedded Market Forecasters (a division of American Technology International, Inc.). The company's in-depth surveys of embedded developers are used by industry, the government, and the military and prime contractors to benchmark developer activities, usages and preferences for benchmarking their own processes and for making decisions regarding software, tools and hardware purchases.

Jerry Krasner, Ph.D., MBA
Embedded Market Forecasters
www.embeddedforecast.com
jerry@embeddedforecast.com
508-881-1850