

# Innovative Development Associates

## *White Paper*

### Eight Steps to Improve Your Product Development Batting Average

---

*By Joseph Kormos*

Principal, Innovative Development Associates

Director, The Product-MASTERS Collaborative



**Innovative  
Development  
Associates**

***Customer Centered Product and Market Development***

# Eight Steps to Improving Your Product Development Batting Average

*By Joseph Kormos,*

*Principal, Innovative Development Associates*

*Director, The Product MASTERS Collaborative*

---

Driven by intense competition and the desire to penetrate new markets, software companies spend 20% of their sales and more on developing new products and enhancing existing ones. Yet, over 40% of new software products fail to achieve their business goals. How can you improve your batting average at delivering products that meet schedule and budgets, offer competitive uniqueness and generate word of mouth market momentum?

The following eight-step program will help you to identify your weak spots and focus attention where it can do the most good in tuning your new product development effectiveness.

## 1. Commit to Change

In most software companies top manager's attention is focused on closing that new account, hiring a salesman or building alternative distribution channels. These issues always seem like the more urgent problems. Product development, because of it's longer focal length, is rarely a solution for your "90 day sales problem". Hence unless a crisis comes up, management attention is diverted to other areas.

But product development, if it is truly the life blood of the company that many claim it to be, needs serious management attention as well. The first step begins when you decide you have had enough of late to market and wrong to market products and you're going give product planning and development the priority it requires.

To convince yourself of the need to tune up your product development machine examine the following issues:

- What portion of your revenue is from new products ( "new" = half the period of your product life cycle). Is it declining as the years go by?
- What percentage of new products have reference-able customers after 6 months?

- How do you evaluate success and failure of new products? How regularly is success achieved?
- How many products have you released in the past 18 months Vs your leading competitor?
- What is the ratio of your typical development cycle to competitors development cycle

If you don't like the answers, or don't have answers - dig in.

## 2. Know Thyself

Before you run off and start to fix your product development system you need to take stock. What are your strengths? What are your opportunities? Assessing your needs includes examining how you link products with strategy, how product ideas are screened and measured, how well organized your development process is and how you build and manage customer relationships.

If you feel you can perform this analysis honestly, openly and expeditiously do it in house. If you feel like it's time for an independent opinion on where to focus your energy consider enlisting the services of an outside specialist who focuses in the area of product development.

A full scale assessment should include the areas of Management, Product and Business Planning, Marketing and Sales, Tools and Methods, Core Product Development Practices and Customer Involvement .

The following are a few of the key issues to examine:

- How well have you articulated a clear product strategy?
- Have career paths for product development professionals been defined and followed? Are your best people encouraged to contribute to product planning, product marketing and development?
- Do you track products and budgets against plans?
- Does your sales compensation system match your product plans
- How do you communicate and share information on product development successes and lessons learned or do you repeat the same mistakes over again.

## 3. Focus on the Fuzzy Front End

One of the immutable laws of product development is the law of the Fuzzy Front End:

*It costs more to make changes at the end of a program than at the beginning  
...but you don't know what you're doing wrong until you're almost done.*

Manufacturing companies have been focusing on this issue for about ten to fifteen years. They often call it "Concurrent Engineering". Manufactured products require expensive tooling to produce and, before a company invests millions of dollars in molds or dies, an executive sign off is required. As a result product developers treat tooling release as a serious early milestone and try to assure themselves that changes will not be required after this tooling commitment.

In software products, early decisions are often taken more lightly. There is no expensive "manufacturing tooling" milestone that forces early decisions. If changes are made late it seems easy and inexpensive to simply "change a few lines of code". Scrapping inventory is less of an issue than it is with manufactured products.

Such laxity can be expensive however. The leading cause of product delay in software products is the late discovery of customer requirements. Executing such changes late in development costs

50-100 times more than when undertaken in design. If the change takes place after the product is released to the market the cost is likely to be 1000 times greater.

To avoid falling prey to the curse of the Fuzzy Front End consider these important issues:

- Examine when does management make inputs and in what form. What is the ratio of management time spent in first one-third of a development program vs. the last one-third? Most companies find that less than 5% of manager's time is applied in the early stages of product development. Work to increase that to at least 20%.
- Evaluate the causes of late changes and delays. Track them back to root causes and take action.

## 4. Define Your Product Realization Process

At many software companies it can be said that "all products are prisoners and occasionally one escapes". Too many software companies operate with a process which was never designed in the first place, is not documented and where exceptions are the rule.

One sure sign of a broken process is incessant bickering about "what the customer really wants". Sales, marketing, planning and customer support look out at the customer through their own silo and see something different - a bit like the blind man and the camel. The specifics are excessively long product definition cycles, alternative requirements documents or none at all, lack of respect, an unwillingness to openly discuss issues and uncontrolled additions which get "thrown into" the product.

As you examine your process look for how well you do in the following areas:

- Do more than 20% of projects proceed to full scale development without a financial study, approved requirements, market size estimate and a clear statement of the product's financial value proposition?
- What proportion of projects are "special cases" and don't follow the standard development process? Has this flexibility really saved time or cost you in inaccurate documentation, poor reliability or ineffective sales training/
- What proportion of the development team makes a clean break after release of a new product? What percent stay behind to deal with customer issues and unfinished business?
- What proportion of features are dropped from the original spec to get the product out the door?

## 5. Less is More

The best expressway system in the world bogs down when too many vehicles are on it. In the short term, the most significant action you can take to improve your product development batting average is to make sure that you are reserving your limited development resources for the programs with the best potential.

The best companies place special emphasis on screening ideas and gates for tracking progress and building consensus. Good ideas advance steadily. Losers are killed and stay dead. More resources are applied to a smaller number of good ideas. Products happen more quickly.

The criteria for evaluating programs need not be elaborate but should be consistent throughout the development process. A basic set of criteria might include a market value screen (Does the customer care?) a business value screen (Do we care?) a competitive screen (Can we win?) and a project execution screen (Can we do it?).

To evaluate how well you perform in this area spend time examining:

- Where do product ideas come from and which sources are most successful.
- Funnel Ratios: The proportion of new product ideas proposed, the number evaluated for feasibility, the number beginning full scale development, the number released to the market and the number achieving their business goals.
- Number of projects undertaken without an explicit link to overall strategy

## 6. Customer Understanding

If third hand, "squeaky wheel" conversations with salesmen who just lost an order are your primary source for setting your development agenda you're probably focusing on the wrong end of the train. There is nothing worse than force feeding a product plan to include a "gotta have it" feature in the name of field responsiveness only be greeted with deadly silence from the marketplace. Invariably the salesman is gone, the customer is gone, the competition has something better and you are left holding the bag for those wasted resources.

Getting out front of the competition requires more than perusing your enhancements database or conducting beta tests. Likewise it is more than conducting a few focus groups or calling up your favorite customer to ask them what they think. To develop customer inspired, as opposed to customer defined, products a variety of methods is needed to capture broad spectrum of input from various user constituencies. (See boxes B & C)

From our research on how software companies plan products, the most rapidly growing customer research method is Customer Visits by cross functional teams. Unfortunately, while most companies conduct customer visits in some form, many cut corners by limiting the number of customers visited, poorly planning the discussions and failing to archive the results for subsequent use. As a result the impact is limited.

To audit your effectiveness at capturing the voice of the customer consider the following issues:

- What proportion of product development effort is spent on understanding the customer and the competition as opposed to designing, coding and testing?
- What is the ratio of marketing/planning resources to full time equivalent developers.
- For your most recent new product, what proportion of the customer requirements changed after the start of full scale development?
- How long does it take to achieve agreement on product definition?
- Are new products often missing services, interfaces or key features needed by customers?

Box A
<b>Customer Research Methods for Software Companies</b>
<b><u>Traditional Methods</u></b>
Customer Enhancement Request Lists
Beta tests
Customer Support Data bases
User group discussions and feedback
Salesman/ Application support person interviews and meetings
Win Loss Analysis
Customer Satisfaction Surveys
Phone Surveys
Focus groups
Mail Surveys
<b><u>Less Traditional Methods</u></b>
Customer Visits: Cross Functional teams
Customer Advisory Boards
Contextual Inquiry
Ethnography/ Unobtrusive Observation/Videography
Lead user discussions and interviews
Cultural Anthropology ... experience the product
Prototypes: paper/ Electronic
Usability labs
Discussions with Competitor's Customers

## 7. Institutionalize Customer Understanding

Collecting information on customer attitudes and needs isn't enough. You have to make it stick. What is asked is as important as who is asked is as important as who hears it, what is remembered and how it's used.

One key to getting customer insight applied effectively is the utilization of structured methods for translating soft customer requirements with measurable design goals. One such method is known as Quality Function Deployment or QFD. The QFD method combines information on customer demands, competitive alternatives and design features which can meet these needs into a matrix or "House of Quality". The method allows the development team to establish specific, benchmarkable targets for the performance of each product feature. Because the relationship of each feature can be tracked back to specific customer needs the design specification is insulated them from constant reinterpretation throughout the development process.

## 8. Continuously Improve

Improving at new products is a never ending proposition. Consider three key areas to keep your new product engine in good running condition:

- Excellent product development companies work to captured lessons learned and avoid repeating mistakes from previous programs.
- Benchmark other similar organizations to learn from their successes and failures.
- Measure your progress over time by tracking the results of individual products as well as aggregate measures of your success. One good measure of overall quality is to track the ratio of sustaining costs (continuing development costs) to revenue for all products.

New products provide software companies with the opportunity to attract new customers and partners, strengthen existing client relationships, build internal excitement and get out front of the competition. A new products game plan which focuses on management commitment, a coordinated development and planning process and a focus on the customer is the key to reaping those benefits.

### Box B

#### Do's and Don'ts for Capturing Customer Understanding

##### Do

- Build customer understanding efforts into the front of each program. Don't wait until you're confronted with a key design or coding question.
- Use a variety of methods consistently
- Understand that you often need to invest in understanding the actual user environment. Your product is probably not used on the phone or in a focus conference room. To deliver what the customer values you'll need to get first hand understanding of interruptions, rules training, hardware configurations, schedules
- Consider multiple views of the customer including:
  - leaders and laggards; easy and demanding; happy and unhappy; "almost had" and "never had"
- Invest time and effort to develop a repeatable method for getting beneath the surface to discover needs and values. Provide training to the organization in methods for understanding the customer.
- Realize that customers really want to help and they are usually flattered to be asked.

##### Don't

- Think that because you collected the information people will use it.
- Believe everything that customers say. They are easily swayed by their view of what is possible or what they saw from a competitor. They will often tell you what they think you want to hear or what today's problem is. They will almost never tell you about the basics.
- Slant the customer's answer to your question by the way you ask it.
- Underestimate the value of getting the customer's input directly to your developers through their participation in customer visits and interviews.
- Focus on gathering a little bit of information about a lot of customers ... instead learn to understand a lot about a few customers
- Confuse information with insight

---

*Joseph Kormos is author of **Product Planning Practices in the Software Industry**, and **Planning Software Products That Sell** both published by Culpepper and Associates. He is also Director of the **Product-MASTERS Collaborative**. His consulting firm, **Innovative Development Associates**, helps high tech companies benchmark and improve product*

**EIGHT STEPS TO IMPROVING YOUR PRODUCT DEVELOPMENT BATTING  
AVERAGE.DOC**

---

*development processes to boost the payoff from their R&D investment. For more information or to order the book contact 513-683-1911 or [www.inndev.com](http://www.inndev.com)*