Identifying Your Organization's Best Practices

David Herron and David Garmus
The David Consulting Group

As organizations strive to improve the design, development, and delivery of software solutions, an effective combination of processes, tools, and methods is critical. Finding the right combination requires careful analysis of the variables that influence gains in productivity and quality. By utilizing industry benchmark data and performance indicators, organizations can collect and analyze their own data to determine the combination of processes, tools, and methods that provide the greatest impact on their performance. This article reports on three client case studies that incorporated various measurement techniques to determine best practices.

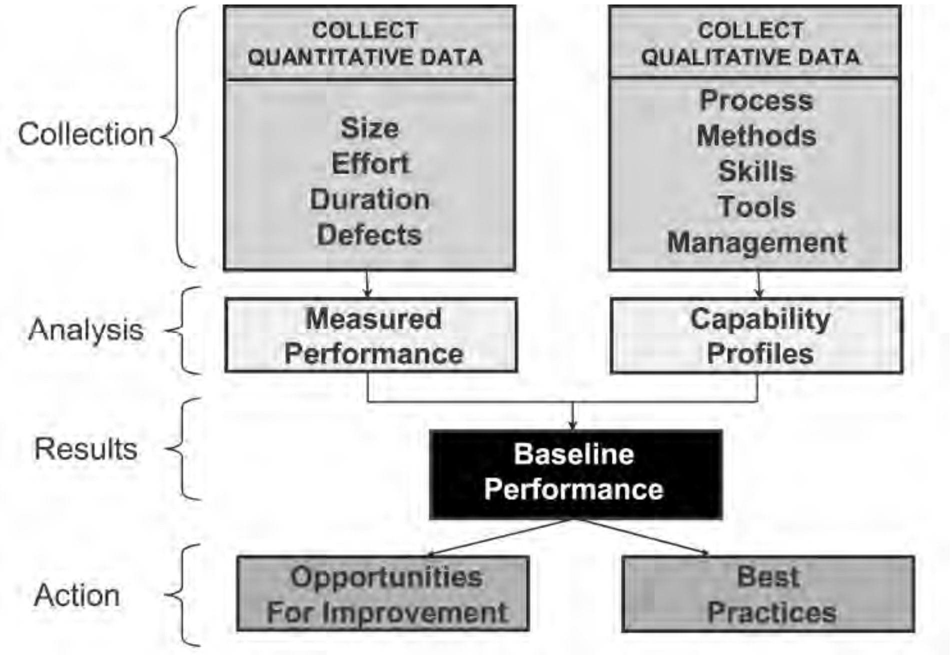
Characterizing an organization's best software development practices can optimally be described as those software development practices that yield favorable results, which are often measured by customer satisfaction, reduced time to market, decreased cost and better product quality. This article will discuss how three different organizations used a combination of quantitative measures and qualitative values to identify their best practices. Based on the knowledge gained, the organizations used the results to improve their development practices and/or to advance their process improvement programs.

In each case, the desire to identify their best practices was driven by senior level management who wanted results that would have a direct impact on stated business goals and objectives. A summary view of their business goals included the following:

- Reduce project costs (mostly labor).
- Improve their time-to-market delivery of software.
- Minimize defects delivered.
- Improve performance relative to industry benchmark data points.

In all three cases (and in many other companies), the organizational strategy to achieve these goals was centered on quick-fix approaches. Cost reduction frequently tops the list and is usually the driving force behind the decision to outsource software development to an offshore provider. Time to market is often reduced by delivering fewer features to the end user, thus reducing the development work load. Defect minimization is too often ignored. We know too well that quick-fix remedies are not usually effective. However, the alternative to achieving sustained and measurable improvement can be a hard pill to swallow. In order to achieve the findings and the results noted in the cases that follow, senior management had a well-defined vision of what they wanted to accomplish and had to marshal the resources necessary to realize the desired results.

Figure 1: *Basic Measurement Model*



The ability to properly set management expectations and to gain their support was enhanced by the introduction of a measurement model that objectively and quantitatively generated meaningful results.

Introducing the Measurement Model

The key to successful performance management is performance measurement. As the software industry grows out of its infancy into a more mature set of practices, the inclusion of performance measurement to manage and direct decisions is becoming more of a mainstream practice. Organizations long ago recognized the need to establish strategic goals and objectives; equally important, however, is the identification of an appropriate set of measures that provide quantitative evidence that those goals and objectives have been achieved.

A basic measurement model that has been advanced by the Practical Software and Systems Measurement program suggests that an organization follow these three steps:

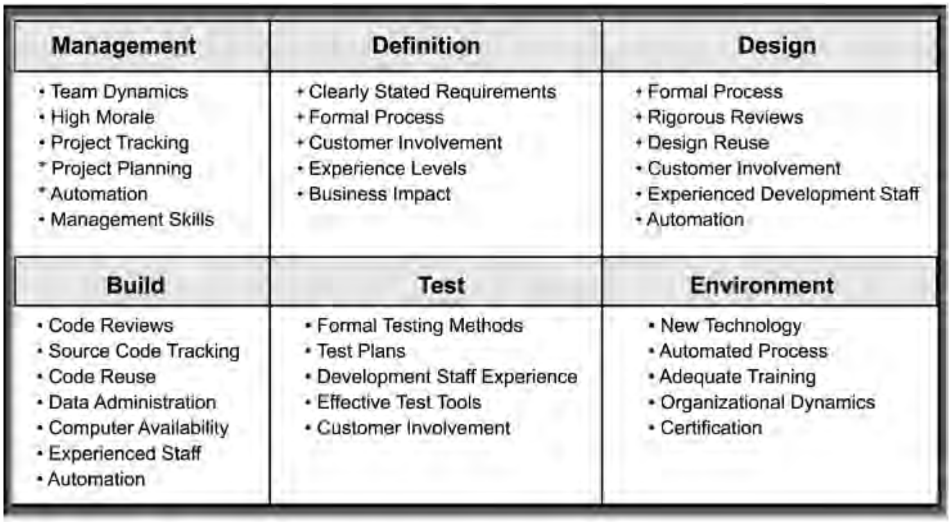
- Identify the needs of the organization.
- Select measures appropriate to measuring whether the needs have been met.
- Integrate measurement into the software development process.

In each of the three cases examined in this article, the management championing the initiative had identified the needs of their organization. Our effort began when management requested our help in selecting the appropriate measures and creating a measurement model that would result in the quantification of process performance levels. Furthermore, we were called upon to utilize a measurement model that would provide the ability to compare internal performance measures to industry benchmark levels of performance.

The basic measurement model we used included the collection and analysis of both quantitative and qualitative elements (see Figure 1). The quantitative elements included four basic measures: size, effort, duration, and defects. The qualitative elements included a variety of data points that were used to evaluate levels of competency regarding process, methods, skills, automation, technology, and management practices.

Collected on a project-by-project basis, quantitative data can be displayed in a measured profile that indicates how well a project is performing. Standard industry measures such as function points (FPs) per person month (PM), defect density and time to market must be calculated. If FPs are used to measure project size, there is an opportunity to make comparisons to industry data points that are also based on FPs.

The qualitative data (again collected on a project-by-project basis) results in a matching capability profile. This profile data identifies the attributes that contribute to high or low yields of performance, such as those indicated in Table 1.



Management	Definition	Design
• Team Dynamics • High Morale • Project Tracking • Project Planning • Automation • Management Skills	• Clearly Stated Requirements • Formal Process • Customer Involvement • Experience Levels • Business Impact	• Formal Process • Rigorous Reviews • Design Reuse • Customer Involvement • Experienced Development Staff • Automation
Build	**Test**	**Environment**
• Code Reviews • Source Code Tracking • Code Reuse • Data Administration • Computer Availability • Experienced Staff • Automation	• Formal Testing Methods • Test Plans • Development Staff Experience • Effective Test Tools • Customer Involvement	• New Technology • Automated Process • Adequate Training • Organizational Dynamics • Certification

Table 1: *Capability Profile Attributes*

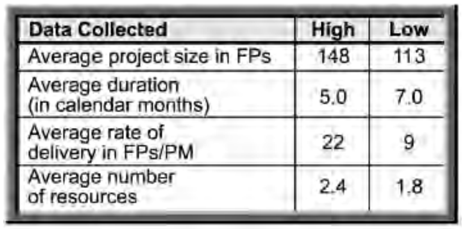
These two elements (quantitative and qualitative) come together to form what is commonly viewed as an organization's baseline of performance. The baseline values are compiled from a selection of measured projects and represent the overall performance level of the organization.

Results vary significantly. Some projects perform very well (i.e., they have low cost and high quality), and other projects do not. The quantitative data provides senior management with an objective view of current performance levels. The qualitative data provides the opportunity to examine the attributes of the projects to determine why certain projects have outperformed others. This analysis effort leads an organization to the identification of their best practices and opportunities for improvement.

The following three case studies used this baseline approach in one form or another. The presentation of the results for each of the case studies varies due to the nature of each unique engagement and how the client wanted the information displayed. There is no magic or silver-bullet discovery (as you will see). Basic measures, applied through a practical baseline model, provided senior management with the information they needed to make a more informed decision.

Case Study 1: Large Financial Institution

Objective: Identify characteristics of high performing projects.
Scope: Conduct an organization-wide baseline study.
Collection and Analysis: Data (quantitative and qualitative) was collected on 65 completed projects. Productivity rates, expressed in terms of FPs per staff month, were calculated along with three other base measures: duration, resource utilization, and cost. The results were divided into two categories (high performing projects and low performing projects), and an average was calculated for each category as indicated in Table 2:



Data Collected	High	Low
Average project size in FPs	148	113
Average duration (in calendar months)	5.0	7.0
Average rate of delivery in FPs/PM	22	9
Average number of resources	2.4	1.8

Table 2: *Case Study 1 Results*

The data demonstrated that high-performing projects produced (on average) more functionality (148 FPs) in a shorter timeframe (five months) with a modest increase in staffing levels to 2.4.

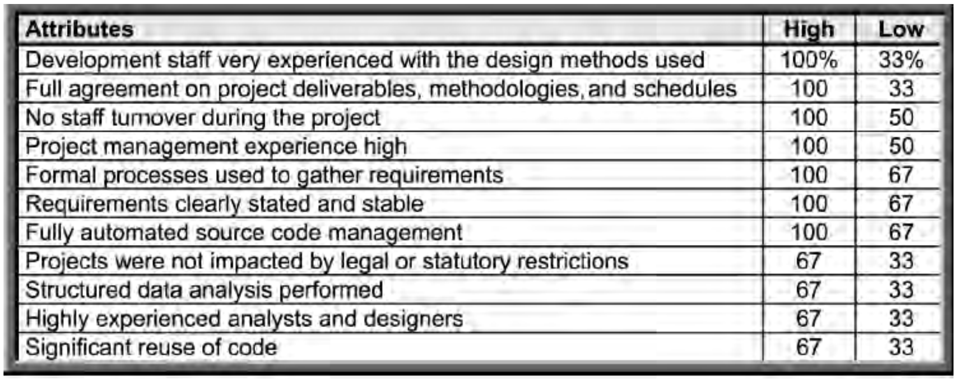
Qualitative data (attributes about each project) was collected, and profiles of performance were developed that identified characteristics consistently present in the higher performing projects, but limited or absent from the lower performing projects. These sets of attributes were then considered to be the leading factors that contributed to higher performing projects.

The findings listed in Table 3 below indicate the attributes and their frequency of occurrence (percent) in the high- and low-performing projects.

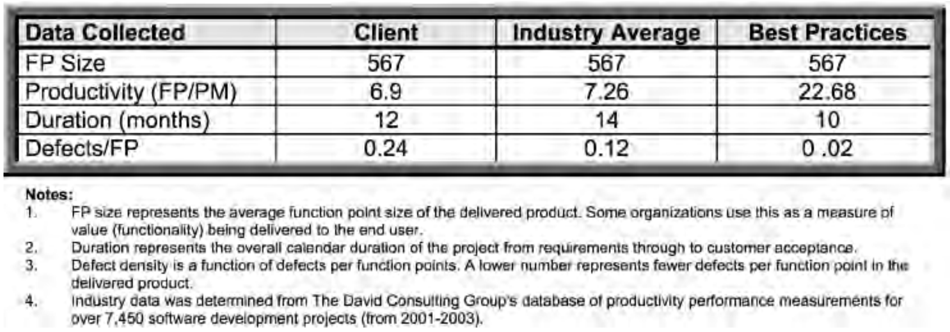
Case Study 2: Midsize Insurance Organization

Objective: Benchmark comparison to industry averages and best practices. Identify best practices opportunities.
Scope: Conduct a baseline study using 25 selected projects.
Collection and Analysis: Measurement baseline data was collected and analyzed to produce performance indicators such as those in the first case study. After deter-

Table 3: *Case Study 1 Findings*



Attributes	High	Low
Development staff very experienced with the design methods used	100%	33%
Full agreement on project deliverables, methodologies, and schedules	100	33
No staff turnover during the project	100	50
Project management experience high	100	50
Formal processes used to gather requirements	100	67
Requirements clearly stated and stable	100	67
Fully automated source code management	100	67
Projects were not impacted by legal or statutory restrictions	67	33
Structured data analysis performed	67	33
Highly experienced analysts and designers	67	33
Significant reuse of code	67	33

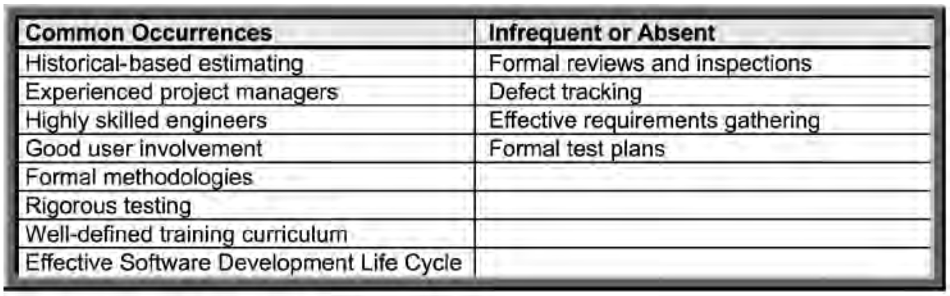


Data Collected	Client	Industry Average	Best Practices
FP Size	567	567	567
Productivity (FP/PM)	6.9	7.26	22.68
Duration (months)	12	14	10
Defects/FP	0.24	0.12	0 .02

Notes:

1. FP size represents the average function point size of the delivered product. Some organizations use this as a measure of value (functionality) being delivered to the end user.
2. Duration represents the overall calendar duration of the project from requirements through to customer acceptance.
3. Defect density is a function of defects per function points. A lower number represents fewer defects per function point in the delivered product.
4. Industry data was determined from The David Consulting Group's database of productivity performance measurements for over 7,450 software development projects (from 2001-2003).

Table 4: *Case Study 2 Performance Indicators Comparison*



Common Occurrences	Infrequent or Absent
Historical-based estimating	Formal reviews and inspections
Experienced project managers	Defect tracking
Highly skilled engineers	Effective requirements gathering
Good user involvement	Formal test plans
Formal methodologies	
Rigorous testing	
Well-defined training curriculum	
Effective Software Development Life Cycle	

Table 5: *Case Study 2 Presence or Absence of Attributes*

mining the current level of performance, a comparison to industry average and industry best practices benchmarks was conducted. The results are shown in Table 4.

We examined these data points and analyzed the underlying profile data. Within this sampling of projects, the client's productivity rate was close to the industry average (6.9 versus 7.26); however, plenty of opportunity for improvement still existed as evidenced by the best practices benchmark. The client was actually delivering products (on average) in a shorter timeframe than industry average, and again there was opportunity to improve as the organization moved towards best practices thresholds. Finally, the level of quality (defect density) was significantly below industry data points.

Looking at the findings in this picture (see Table 5), we observed an organization that was getting their software product out the door quickly by increasing staffing levels and shortcutting quality practices. This was further substantiated by evaluating the attributes that were the most common and those that were the most conspicuous by their absence.

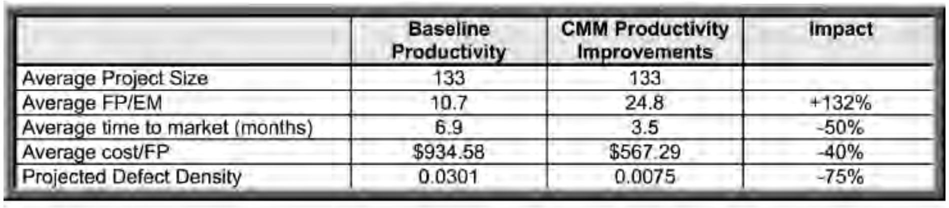
These common occurrences in Table 5 refer to practices that the client was already executing. Our analysis suggested that if the client were to focus on the infrequent or absent practices noted in Table 5, they would see a substantial improvement in their level of quality without sacrificing productivity or duration.

Case Study 3: Large Service Organization

Objective: Identify impact of moving to the Software Engineering Institute's Capability Maturity Model® (CMM®) Level 3.
Scope: Perform baseline measures on a sample set of representative projects.
Collection and Analysis: The final case study involved an organization that wanted to estimate the impact that a CMM Level 3 process improvement initiative would have on their performance. They attributed the process areas associated with CMM Level 3 to best practices. To model this improvement, the organization had to first determine its current baseline of performance and establish a composite profile of contributing attributes.

Project data was again collected and analyzed. Averages for size (FPs), productivity (FPs per effort month [EM]), duration (calendar months), and cost (labor) were computed. Using a composite profile, a mapping of the current project attributes for the organization was developed. In parallel, another model was developed for projects of a similar size with a mapping of attributes that matched a CMM Level 3 organization. A modeling tool – Predictor from DDB Software, Inc. – was used to accomplish the modeling. Predictor contains a series of algorithms that are used to calculate productivity levels such as those noted in the findings in Table 6 for the CMM productivity improvements. The values within Predictor are based upon the statistical analysis of software process attributes from more than 8,700 client projects.

Table 6: *Case Study 3 Productivity Levels*



	Baseline Productivity	CMM Productivity Improvements	Impact
Average Project Size	133	133	
Average FP/EM	10.7	24.8	+132%
Average time to market (months)	6.9	3.5	-50%
Average cost/FP	$934.58	$567.29	-40%
Projected Defect Density	0.0301	0.0075	-75%

The projected impact of CMM Level 3 practices for this organization was significant. For the same size project, productivity (FP/EM) was projected to increase by 132 percent, time-to-market reduced by 50 percent, cost reduction by 40 percent and defect density reduced by 75 percent. This modeling technique helped this organization evaluate the potential benefits of CMM process improvement.

The potential impact indicated above may appear to be dramatic, but that is a matter of perspective. Certainly, this significant gain in productivity and reduction in defects would exceed most expectations; however, if the baseline productivity were dramatically below industry averages based on the nature of the process profile, then clearly large gains could and should be expected.

In Summary

These three case studies exhibit a variety of ways in which measurement data can be used to learn more about the following:

- An organization's level of performance.
- Key factors that contribute to high or low productivity yields.
- The level of performance as compared to industry data points.
- The potential impact of strategic initiatives through the use of performance modeling.

Utilizing a measurement model that includes both a quantitative perspective and a qualitative perspective is most important. It is from this vantage point that an organization can access both the measured performance profiles along with an understanding of the process profile elements that contributed to the results. The process profiles have the added advantage of recommending a

® Capability Maturity Model and CMM are registered in the U.S. Patent and Trademark Office by Carnegie Mellon University.

direction for future improvement strategies.

In reviewing the experiences and results from these three client case studies, readers should not assume that similar outcomes would be achieved in their organizations. The prudent action would be to take your own measures and create your own organizational performance baseline. Utilizing industry-accepted measures such as FPs will allow you to perform the necessary comparative analysis. The investment in a baseline study is relatively insignificant in comparison to the value of the information gained. Of course, the return on that investment can be realized only with the proper execution of improved software development practices.◆

About the Authors

David Herron is co-principal and co-founder of The David Consulting Group. He is an acknowledged authority in using metrics to help organizations monitor the impact of Information Technology (IT), the advancement of IT organizations to higher levels on the Software Engineering Institute Capability Maturity Model®, and the governance of outsourcing arrangements. Herron assists clients in establishing software measurement, process improvement, and quality programs and to enhance their project management techniques. He has more than 25 years experience in managing, developing, and maintaining computer software systems. Herron serves as a Cutter Consortium Expert Consultant and is past chair of the International Function Point Users Group (IFPUG) Management Reporting Committee, a member of the IFPUG IT Performance Committee, and a member of the American Society for Quality. He is a co-author of "Measuring the Software Process: A Practical Guide to Functional Measurement," and "Function Point Analysis: Measurement Practices for Successful Software Projects," and has contributed numerous articles to industry publications and lectured worldwide on functional measures. He attended Union College and Northeastern University.

The David Consulting Group
19 Point View DR
Medford, NJ 08055
Phone: (609) 654-6227
Fax: (609) 654-2338
E-mail: dcgherron@comcast.net

David Garmus is co-principal and co-founder of The David Consulting Group. He is an acknowledged authority in the sizing, measurement, and estimation of software application development and maintenance. He has more than 25 years of experience in managing, developing, and maintaining computer software systems. Concurrently, he served as a university instructor in computer programming, system development, information systems management, data processing, accounting, finance, and banking. Garmus is past president of the International Function Point Users Group (IFPUG) and a member of the Counting Practices Committee. He previously served IFPUG as chair of the Certification Committee, as chair of the New Environments Committee, and on the Board of Directors as director of Applied Programs and vice president. Garmus is co-author of "Measuring the Software Process: A Practical Guide to Functional Measurement," and "Function Point Analysis: Measurement Practices for Successful Software Projects," and has contributed numerous articles to industry publications and lectured worldwide on functional measures. He has a Bachelor of Science from the University of California Los Angeles and a master's degree in business administration from Harvard University.

The David Consulting Group
1935 Salt Myrtle LN
Orange Park, FL 32003
Phone: (904) 269-0211
Fax: (904) 215-0444
E-mail: dcg_dg@comcast.net