Decision Support for Software Management

S. Chulani, P. Santhanam, R. Leszkowicz
{sunita, pasanth, bobl}@us.ibm.com

Abstract:

Typical software metrics and measurements (e.g. code coverage, McCabe complexity, OO metrics, etc) target the technical professionals in a software development organization. While this is useful to the practitioners in the trenches, the managers and executives of these organizations face a broader set of issues. In order to make better decisions, the collection of business and software engineering data and its regular use throughout the product life cycle is essential. The product life cycle typically consists of requirements, design and development, service and support during which various pieces of information are collected. This paper presents (i) examples of data and the metrics currently available in a typical large software organization (ii) highlights some of the issues in integrating these data sources and the lessons learned (iii) analyzes the data and presents models that enable management to derive relationships among the inhibitors and drivers of these metrics and (iv) presents a framework for a decision support system.

Keywords: decision support, software metrics, knowledge management, and information integration.

1. Introduction:

One of the key distinguishing factors of a successful business is its ability to sense change earlier and respond to the change quicker than its competition [1]. In a software organization, the adaptive loop of sensing change early, making decisions to improve the business and acting on those decisions can be tremendously enhanced by the use of software metrics. Numerous insights are developed from the rich, in-depth data gathered on an ongoing basis throughout the different activities of the software product life cycle. Often, this data needs to be understood and transformed into useful information in the correct context.

The software product life cycle typically consists of a series of activities: requirements, design and development of the product, service and support of the product. During each of these activities, detailed data on quality-related metrics, ranging from in-process development metrics to post-GA (generally-available) service metrics to customer satisfaction data, are collected with an effort to evaluate and measure software quality. These measurements are essential to improving existing processes, perfecting methodologies over time, and gauging future software projects. All of this information should be integrated in a way to derive the inhibitors and drivers that interplay within the entire product life cycle [2 - 9]. As a result of this linkage, one should be able to successfully predict the implication that an increase or decrease in a certain metric would have on the rest of the metrics within development, service, customer satisfaction and finally, revenue.

2. Data:

The product development teams are the primary producers of the product; the service teams work with the customers once the product is generally available and help the customer if the customer experiences any problems with the product; the customer satisfaction analysts capture and analyze several different marketing aspects of the organization and products such as brand name satisfaction, trends in the marketplace with respect to product customer satisfaction and its relationship with competitor products' customer satisfaction, etc. Often times, these different groups are separate entities with different focus areas. This paper focuses on a subset of the complete product life cycle using data from three separate data sources: (i) customer support, (ii) critical situations involving customers and (iii) customer satisfaction surveys[1]. Each of these data sources is briefly described in the next few paragraphs.

2.1 Customer Support

When customers experience a problem with a product, they call in for support and resolution of their problem. The service teams interact directly with these customers who are experiencing problems with their software or seeking information for problem resolution. The service teams populate a lot of data including the nature of the problem, its severity, problem resolution provided, etc. in the customer support and service database.

For example, when a customer calls in with a problem, the service representative opens a 'Problem Management Report' or more commonly known as a PMR at IBM. The service representative then works with the customer and/or the development team to resolve the problem. The resolution of the problem may or may not cause a change in the product. If a change is made, the problem is termed as a 'Defect Oriented Problem' and is coded as an 'Authorized Program Analysis Report' or an APAR. The change is a PTF or a 'Program Temporary Fix' until it is incorporated in the next release. If a change is not made, the problem is a 'Non-Defect Oriented Problem' and could be related to difficulty in installation or usability, etc. The process is illustrated in figure 1.

[1] Note: This paper that does not capture all the processes/analyses done in a large organization. The field of customer satisfaction and software quality is a broad and complex field and this paper makes an attempt to uncover some useful insights that would help management make useful decisions.

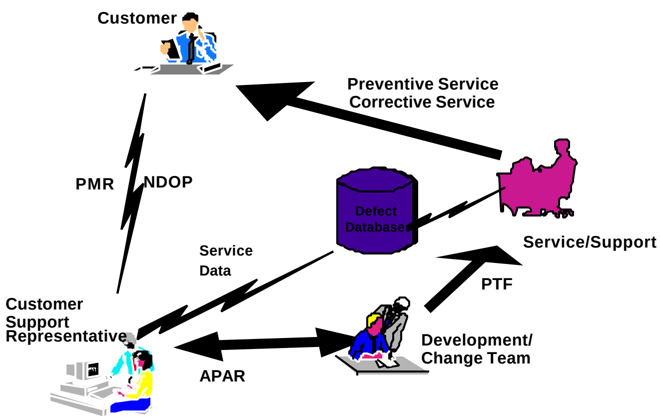


Figure 1: Customer Support at IBM

2.2 Critical situations involving customers

Sometimes, customers experience high impact and/or high severity problems that need immediate attention and resolution and often involve very senior management. Detailed data for these critical situations is gathered for tracking and future analysis. Critical situations are given the highest priority for resolution as they often involve appreciable losses and outages to the customer. A developer (or a team of developers) may work onsite with the customer to expedite the resolution process. The high impact/urgent nature of these problems warrant for a different tracking mechanism as compared to the regular customer support problems.

2.3 Customer satisfaction surveys

The customer satisfaction process starts with the identification of customer communications and issues regarding IBM software. These communications are identified through surveys, complaints, customer meetings, user groups or software industry consultant studies. From this source of customer inputs, IBM representatives take appropriate actions, either to address problems or modify current marketing programs, brand sales campaigns and coverage models.

Data on satisfaction with the Capability, Ease of Use, Performance, Reliability, Ease of Installation, Maintainability, Documentation, Service/Support, Overall Satisfaction, etc. of the product (popularly known as the CUPRIMDSO measures) is collected to determine if the product is meeting customer expectations. Customer Satisfaction targets are based on comparing year-to-year improvements within same business units as well as comparing to other competitors worldwide.

For each of the dimensions, a qualitative scale is used which has ratings such as (i) Very Dissatisfied (ii) Dissatisfied (iii) Neutral (iv) Satisfied (v) Very Satisfied. The Net Satisfaction Index is computed using the responses on the rating scale as a weighted satisfaction index. The data is collected by regularly surveying the customer base. In addition, some unstructured data to understand customer's expectations is also gathered.

3. Information Integration:

All the data described above is collected and maintained by different groups within a typical software organization. The data is not necessarily tracked at the product level as products are composed of components, making it easier to resolve problems at that component level. In the context of the above data sources, the customer support data is at the component level and the critical situations and customer satisfaction data is at the product level. Figure 2 visualizes the challenges in establishing the links between the 3 data sources to integrate them at the product level.

In the customer support data, a product may be represented in several ways: component ID, component name, Product ID (PID) name and PID number. There is a one-to-one correspondence between a PID name and a PID number, and one-to-many correspondence between a component ID and a component name, but a many-to-many mapping between PID number and component ID (since several components could make up a product.)

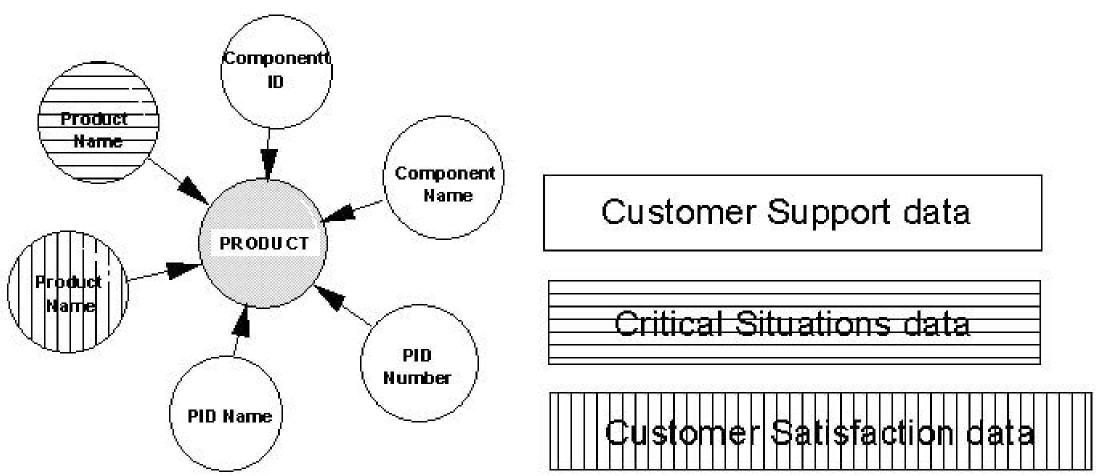


Figure 2: Product Identifiers in the 3 data sources.

The critical situations and customer satisfaction data sources represent products by product name. Due to historical reasons the product names in each of the data sources vary significantly from each other. As a result, mapping the data at the product level in the three data sources is the first step to enable further product level analytics.

Some other issues and roadblocks that we need to resolve to integrate the 3 data sources and build the decision support system are outlined below:

- **Availability of customer information -** This issue is similar to the problem described above with the product identifiers. In order to resolve problems in a way to satisfy customers, it would be extremely beneficial to connect the customer information available in the various data repositories. As an example, let's consider a customer who attempted to use the regular service and support organization to resolve a problem they were having with the software product. The call this customer initially made was recorded in the customer support database as a PMR, possibly an APAR. If the regular service and support organization did not resolve this customer's problem, this customer may choose to submit a complaint and may want to escalate it to a critical situation if the problem is severely impacting the customer's business or relationship with IBM. Although, there exists a field within the critical situations data to link this complaint to a PMR, it is not always filled in. As a result, it is very often impossible to map the customer complaint to the PMR. Moreover, there is potential to link these customer situations to responses from the customer satisfaction survey. For example, if a customer reports a problem, we can explore what satisfaction rating this particular customer gave for the various CUPRIMDSO attributes and how this is reflected in the type of problem they are experiencing. More specifically, we could see if the problem is related to the performance of the software product, is this information reflected in the survey response given by the customers.
- **Limitations imposed by the database design –** The database design of the original databases limits certain types of analysis. The structure of the information in data warehouses helps in answering certain kinds of questions [1]. For example, time lag analysis linking the impact of certain improvements in service to customer satisfaction is hard to do due to the differences in the designs of the two databases. Replication schedules vary for these databases because of the type of information they contain. The critical situations data needs to be updated on a daily basis because customer complaints, especially critical situations, need to be monitored on an ongoing basis. The customer satisfaction data gets refreshed with new data monthly. The tables in the customer support database get refreshed on a daily, weekly or monthly basis, depending on the type of data they contain. The different schedules associated with the many tables pose difficulties in doing time series or time lag analyses.
- **Limitations in analyzing structured and unstructured information together –** A lot of the quality data in these data sources is a mix of structured and unstructured information. Let's take customer satisfaction data as a primary example. The database contains survey data, which includes ratings for various customer satisfaction attributes relating to a software product, as well as, open-ended text questions, which allow the respondent to comment on why they gave a particular score. In order to make valuable use of the respondent comments, one must be able to analyze this data. The other two data sources present a similar problem.

While all three databases contain information such as scores, counts and situation status, there is a lot more value to be derived from mining the written comments. When a customer submits a complaint, the support personnel record a lot of valuable narrative text about a situation. These comments could give a lot of insight as to why or how a particular situation came to exist and how it may be satisfactorily resolved. Recent advancements in text analysis tools improve our ability to derive useful information from text data but they need considerable domain-specific tailoring to maximize their results.

The aim of the project described in this paper is to develop a central warehouse that provides one common repository for all this data. Many of the issues outlined above will be addressed to enable better decision support and management of our software products.

4. Design and Architecture of Decision Support System:

The data described in section 2 is integrated in the single data warehouse at the product level to enable building models and analytical decision support that management can use to exploit the various kinds of data available. Figure 3 presents the high level architecture of the decision support system. As illustrated, the 3 data sources on the different host systems are integrated into a single data warehouse. To ensure controlled access to the data appropriate security standards are deployed. An analytical engine is built on top of this data warehouse to analyze the data from each of the data sources at the product level to facilitate better decision support. The analysis is then presented in the form of a web front-end which represents the quality dashboard. A lot of focus is being given to the design and implementation of the user interface to ensure the successful deployment of this dashboard to senior executives and management. This decision support system allows users easy access to the data they need to make critical business decisions.

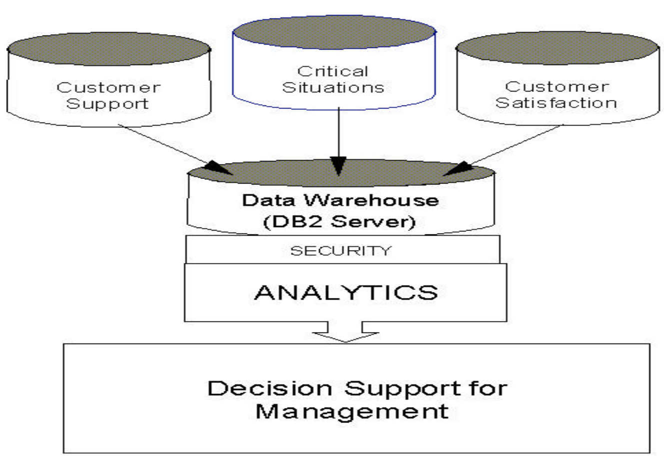


Figure 3: Design and Architecture of Decision Support System

5. Analytics of the Decision Support System:

Once the integrated data is available as described in section 4 and overcoming some of the issues described in section 3, the warehouse can be used to successfully understand the sensitivity of the metrics across development, service, customer satisfaction and finally, revenue. This section presents several decision support examples of analysis that help an organization sense and respond to their customers [1, 10, 11]. *To make the value proposition clear and the analysis easy to understand in the following examples of section 5, we use actual metrics (i.e. attributes) but not actual data.*

5.1 Customer View of Quality from Analysis of Unstructured Information

This subsection presents analysis to answer the question: what are the top quality attributes customers care about the most?

One question in the customer satisfaction survey is: "What quality characteristics come to mind when considering a high quality software product on this operating system?" Answers to this question are in the form of free text. Sometimes the respondents have a long list of characteristics and sometimes just a single attribute. This data gives us the most direct response from our customers to improve customer satisfaction. Using this data and text mining on the free text answers we can understand what quality characteristics are important to our customer and what we should focus our efforts on. Such analysis may reveal that in addition to CUPRIMDS, other attributes such as Speed, Compatibility, Price, etc., may also be relevant. Figure 4 presents analysis done on this open-ended text question. We classify the text responses from our customers based on attributes and then plot a histogram of these attributes. Domain-specific knowledge is required to determine the attributes and their synonyms. From figure 4, it is clear that R, U and P of the CUPRIMDS measures are the top 3 attributes that our customers care about the most. In addition, other attributes such as Speed, Availability, and Integration have a higher significance than Maintainability (M component of CUPRIMDS).

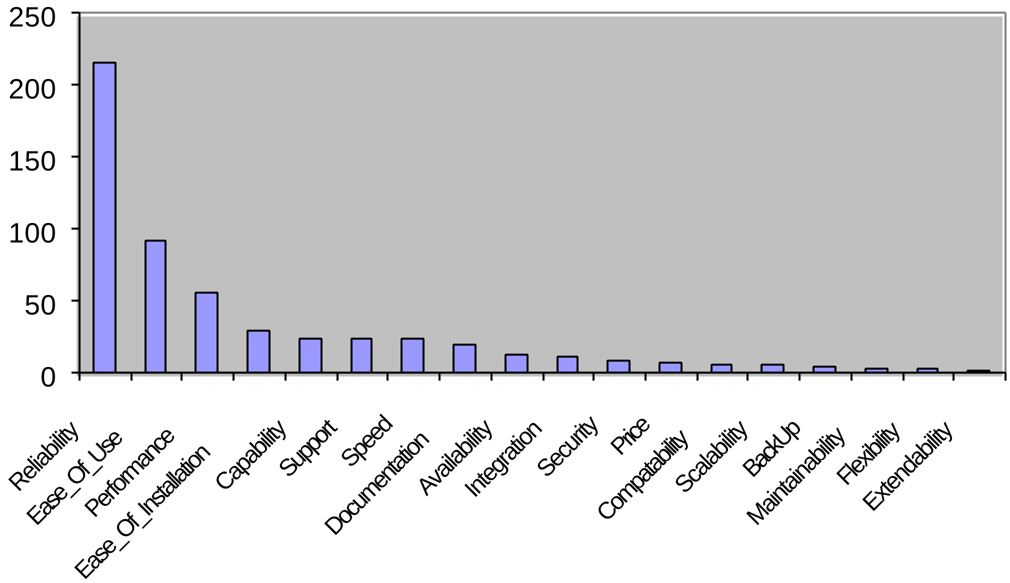


Figure 4: Top three quality attributes: Reliability, Ease of Use, and Performance.

5.2 Managing Investment through Gap Analysis

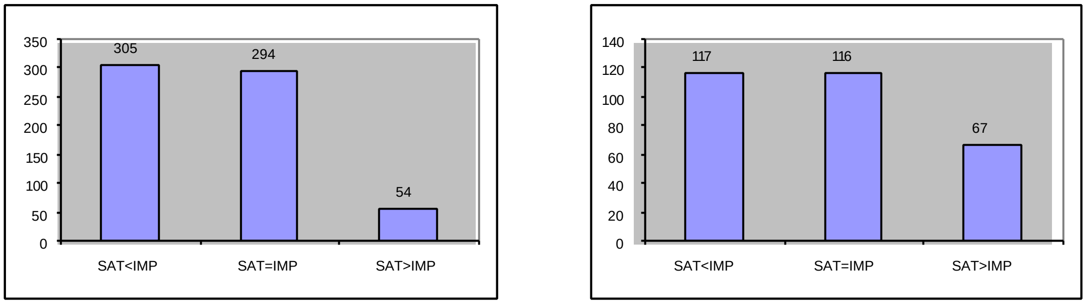
For each of the CUPRIMDS attributes we ask our customers two questions: (i) How satisfied are you with the attribute for this product (the response is a SAT rating on a scale of 1-5) and (ii) how important is this attribute (the response is an IMP rating on a scale of 1-5). This data can help us do gap analysis to understand if we are meeting our customers expectations. The data can be categorized for each respondent as follows:

SAT<IMP - not good as we're not meeting our customers expectations

SAT=IMP - exactly meeting our customer's expectations

SAT>IMP - exceeding our customer's expectations; good but we could use our resources in a better way probably trying to improve other things - maybe to further improve the top 3 attributes - Reliability, Ease of Use and Performance

Let us now focus our gap analysis on two attributes, Reliability and Maintainability, as depicted in figure 5.



Reliability Maintainability

Figure 5: Gap analysis.

For both these attributes, the number of responses for SAT<IMP is far greater than the number of responses for SAT>IMP. This means that for more than half our customers we're not meeting their expectations. That is, our data distribution indicates that our customers are not getting the same level of Satisfaction, as they think is Important. From section 5.1, we already know from the text analysis of the open-ended question that Reliability is amongst the top 3 attributes whereas Maintainability is not that important to our customers. Hence we can now focus all our efforts on improving Reliability i.e. we really need to focus our investments in reducing the height of the SAT<IMP bar for Reliability and other high impact attributes

5.3 Identifying Potential Actions via Correlation Analysis

This section helps us understand the actions we can take earlier in the life of the product to improve the overall quality and customer satisfaction For example, what can we do in service to improve overall customer satisfaction with our products?

When a customer has a problem with a software product they contact our service teams who help resolve the problem. As described in section 2, in the service process, a lot of detailed information about the problems encountered in the field is recorded. If we can analyze the service data to understand which metrics are most highly correlated to Reliability then we have a mechanism of focusing on the improvement of those metrics earlier in the life cycle for the next release of the product to have a positive impact on satisfaction with Reliability [4, 5]. This process is depicted in figure 6. The left side of the figure illustrates the current situation of the data sources in isolation. The right side of the figure shows the relationships being established between service and customer satisfaction to understand how a change in service impacts customer satisfaction.

Doing a correlation study on our data, we find that Reliability is most highly correlated to PMRs of severity 1 (i.e. field problems reported by customers with the highest severity) and Time to Resolution (TTR i.e. the average number of days it takes to provide the resolution to the problem). Hence, focusing on improving TTR of PMRs of severity 1 will definitely have an improved impact on customer's satisfaction with Reliability. Further analysis also shows that just reducing the number of PMRs of severity 1 or just improving the TTR doesn't cause an improvement in Reliability's satisfaction rating. So, a focused improvement in TTR of these PMRs is needed. Similar analysis with the other metrics in the customer support and critical situations data sources establishes further relationships that can be exploited for improving the quality of the software product.

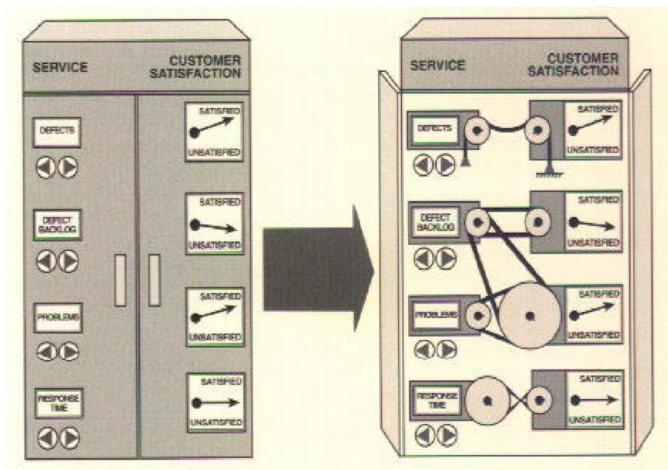


Figure 6: Developing relationships between service and customer satisfaction.

Several other analyses have been done using the service metrics, complaints and critical situations data and customer satisfaction data to improve quality and customer satisfaction of our products. Such analysis helps us answer the questions posed at the beginning of this section.

The analytical engine of the decision support system analyses the inherent dependencies between the pieces of information gathered at the various stages of the life cycle. This gives the decision makers the capability to quantitatively understand how a change in one piece of the model will affect the remaining parts, and therefore, gives them the ability of making better-informed business decisions.

6. Conclusions:

Managers in software development organizations need to make informed decisions in areas such as effectiveness of customer support, meeting customer expectations, increasing market share, etc. This paper illustrated how we can make effective use of existing data in IBM to meet the demands of a fast-changing business. We showed how existing processes can be improved by integrating available data and providing a tool to facilitate better decision-making.

We described different types of data that are gathered during the software product life cycle and some of the challenges faced in integrating and analyzing the data to improve the quality and customer satisfaction of a product. We illustrated that the collection of business and software engineering data and its regular use throughout the product life cycle is essential. The decision support tool described in this paper represents a step forward in making real time assessments about quality based on wide variety of relevant data. In addition, as new data becomes available, the real time analysis tool can be dynamically updated to develop an optimized set of decision support functions. We hope this is a step towards managing a fast paced software business in the 21st century.

7. References:

[1] Haechel S., "Adaptive Enterprise: Creating and Leading Sense-and-Respond Organizations", HBS Press, 1999.

[2] Boehm, B.W., Abts, C., Brown, A.W., Chulani, S., Clark, B.K., Horowitz, E., Madachy, R., Reifer, D. and Steece, B., "Software Cost Estimation with COCOMO II", Prentice-Hall, 2000.

[3] Putnam, L.H. and Myers, W., "Measures for Excellence", Yourdon Press Computing Series, 1992.

[4] Chulani, S., Santhanam P., Moore D., Leszkowicz B., Davidson G., "Deriving a Software Quality View from Customer Satisfaction and Service Data', European Software Conference on Metrics and Measurement, Mar 2001.

[5] Buckley, M. and Chillarege, R., "Discovering relationships between service and customer satisfaction", Proceedings of the International Conference on Software Maintenance, Opio (Nice), France, Oct 1995.

[6] Kan, S., "Metrics and Models in Software Quality Engineering", Addison-Wesley, 1996.

[7] Mendonca, M, "An Approach to Improving Existing Measurement Frameworks in Software Development Organizations", Ph.D. Dissertation, University of Maryland, Computer Science Department, 1997.

[8] Shaw, J., "Customer Inspired Quality: Looking Backward through the Telescope", Jossey Bass Inc., Aug 1996.

[9] Chakrapani, C., "How to Measure Service Quality and Customer Satisfaction", American Marketing Association, 1997.

[10] Marakas, G., "Decision Support Systems in the 21st Century", Prentice Hall, 1998

[11] Turban, E. and Aronson, J., "Decision Support Systems and Intelligent Systems", Prentice Hall, 2000.

About the Authors:

Sunita Chulani
Contact info: Email: sunita@us.ibm.com; Address: IBM Research, 650 Harry Road, B1-226, San Jose, CA 95120

Sunita Chulani is a Research Staff Member at IBM's T.J. Watson Research Center. Her research interests include software metrics and measurement, information integration, knowledge management, software estimation and software process-related issues.

Dr. Chulani received her Bachelors degree in Computer Science from Bombay University and her Masters and Doctorate degrees in Computer Science from the University of Southern California. She has several journal, conference and book publications and has also received two awards from the ISPA (International Society of Parametric Analysts) society.

Padmanabhan Santhanam
Contact info: Email: pasanth@us.ibm.com; Address: IBM T.J. Watson Research Center, 19 Skyline Drive, Hawthorne, NY 10532

Padmanabhan Santhanam holds a B.Sc. from University of Madras, India, an M.Sc. from Indian Institute of Technology, Madras, an M.A. from Hunter College of the CUNY and a Ph.D. from Yale University. He joined IBM Research in 1985 and has been with the Center of Software Engineering since 1993, which he currently manages. His interests include software metrics, structure based testing algorithms, automation of test generation, and realistic modeling of processes in software development and service. Dr. Santhanam is a member of the ACM and a senior member of the IEEE.

Robert J. Leszkowicz
Contact info: Email: bobl@us.ibm.com; Address: IBM Corporation, Route 100, Somers, NY 10589

Mr. Leszkowicz is Worldwide Manager of Quality and Customer Satisfaction Programs for IBM Middleware and Application Software, IBM Marketing and Sales Executive Leader for IBM/Lotus Development Corporation Acquisition Team, Market Planning Manager for IBM Office Systems Software, IBM Systems Engineering Manager.