WebSphere® software

IBM

Redbooks Paper

Jorge Diaz

Improving the WebSphere Support Process

Purpose and scope

This Redpaper provides recommendations for improving the support strategy for different WebSphere® development groups within an organization. It is largely based on experiences gathered from delivering and implementing such a requirement with large corporations across various industries in several countries. The target for these recommendations are medium to large organizations (those that provide support to several groups), although smaller organizations will benefit as well from following some of the items discussed.

The approach described in this Redpaper includes proposing a way to structure the WebSphere support group, an explanation of the different personnel roles involved, how they fit into the recommended structure, and suggestions on raising WebSphere technical expertise throughout the organization. Examples of how this approach has been implemented in other large enterprises, and descriptions of what worked (as well as what did not work) are also given.

This document should be used as an aid in helping to craft the strategy, not as a description of the strategy itself (although it might be fairly similar). Note that this document is not specific to one platform, as we are not describing support for the actual WebSphere runtime operations; that is usually managed by a separate group (especially in historically more tightly-controlled platforms such as zSeries®). The focus of this document is on *application* support.

Overview

There are three stages clearly defined in adopting the support model:

- Phase 1 - identify the roles needed for a pragmatic support group.
- Phase 2 - define the physical nature of the support group.
- Phase 3 - implement and strengthen the group.

These phases are explained in subsequent sections in this document. Other related, important topics are expanded as well.

> **Note:** We do not suggest a “one-size-fits-all” approach, because the internal dynamics of companies can differ greatly. In some situations it might only be feasible to implement one or two characteristics of the strategies presented here. However, adopting the entire approach has been found to provide by far the most positive results.

Rationale

As WebSphere becomes more widely used, and more and more application development groups begin to base their solutions on WebSphere, the need for establishing a focused, effective, and dependable support infrastructure becomes apparent.

In a typical legacy-based enterprise, when a new technology is introduced into an environment, it is done very slowly, usually under the auspices of a small group of supporters.

As the technology becomes accepted throughout the company, more and more development groups utilize it to solve business conundrums requested by different business units. If the technology becomes widely utilized (either by mandate or because of its own popularity), the small group of cohorts that provided support during the starting phases of the adoption process could become overwhelmed rather quickly.

This could lead to development groups becoming frustrated with the technology because they would not be receiving the right level of internal support. If mentors are not available, developers might feel that they have to “reinvent the wheel” at every step. The process of developing a solution can become more and more inefficient, almost to the point of not being practical. Development groups could start lose heart regarding the technology, perhaps complaining to management that “things used to be so much better in the past”. The group that introduced the technology may revisit their decision to accept it, perhaps thinking that it does not work well in their specific environment.

In a worst case scenario, the technology could be removed from the strategic plans. Then either the process starts all over again with another new technology, or the company reverts to the ways it used to develop applications.

All of this churn can be avoided if proper planning and action is given to the *support* aspect of introducing a new technological paradigm. Assuming that the technology provides tangible benefits to the enterprise and that technically speaking, it does improve the way applications are produced, a proper support infrastructure can act as a keystone in the success or failure of the chosen technology.

Background

The support structure suggested in this Redpaper is a WebSphere Support Engineering Team. This support structure provides a one-stop area for “everything WebSphere”: it supplies developers with much needed application design services, and provides standards, recommendations on reuse, directions on quality assurance, and upcoming features.The team includes a management aspect, to ensure all entities work cohesively. This group would represent the WebSphere center of knowledge within the company.

Before describing a recommended support structure, in the following section we analyze what other enterprises utilizing WebSphere for z/OS® and distributed platforms have crafted in this area, and examine the lessons learned in each case.

Case studies

These entities, which are at different levels of maturity, provide useful examples of the intricacies and challenges of providing for a pragmatic, effective support system.

Large European bank

The support structure at this bank is called the WebSphere Engineering office. The term “engineering” identifies the intention of the group, which is to serve as the provider of reliable, pragmatic, fact-based WebSphere information.

The group consists of individuals who provide consulting support to any application group that solicits it. There is also a cluster of people who test new versions of the software, together with any released patches, and serve as the gatekeepers for software to continue forward towards the production environments; they also provide education, as needed.

The group includes some individuals focused on different platforms (specifically Windows®, Solaris, and z/OS) and as a result, the system administration skills required are very different for each case. The number of people involved varies between five and ten. There are also IBM® representatives onsite, full time, to provide extra, direct support if needed.

Lesson: *A separate group seems to produce better results in a big, cross-divisional context.* If the group was a virtual group, seeded with people working for the different divisions involved, it might be harder to achieve the high levels of communication needed for effective knowledge transfer and intellectual capital management.; one person could end up “reinventing the wheel” to solve a problem that someone else had already solved.

Large American retail company

This high-throughput, e-commerce giant gathered together a WebSphere support group under the auspices of IBM. The WebSphere center of excellence was staffed by five IBM consultants, together with five company employees. The group consisted of a program manager, system administrators, service consultants and J2EE architects.

The IBM consultants had a minimum tenure of about nine months, after which it was expected that support would be run fully in-house, with the occasional IBM contact or consulting engagement. This structure worked very well; it was flexible enough to adjust to the ever-changing nature of the business, and provided the in-house talent with a gradual, efficient way to bring their WebSphere skills up to par with the task at hand.

Lesson: *This setup highlighted the need for a program manager, to shield technologists from non-technical issues.* Non-technical issues tend to require a larger investment of time than one would expect, involving following up on critical issues with support and management, delivery of status, and overall direction and planning. This last item is very important, because technical specialists often tend to have a focused approach to solutions, and do not always concentrate on the “bigger picture”. This approach also provided a good test of the pragmatic size for the core of a support group, and proved to be a great way to reduce risk while remaining fully productive.

Large American insurance company

The WebSphere support organization in this case was very typical in that it consisted of several system administrators that knew the runtime, together with several experienced J2EE architects leading the design efforts and several developers new to Java™. The architects and the system administrator were in different departments.

The company provided the same support that it gave to all new technologies. This setup allowed for more control over what was developed and deployed, but did not allow for the creation of an appropriate infrastructure for the future (the system administrators would get easily overwhelmed if additional applications needed their services). Fortunately, the release plans for the company's' core applications spanned several iterations over several years.

Lesson: *Experience in J2EE design helped to minimize the risk with the production application, even though the support structure was not mature enough.* The application was stable and performed its duties as planned.

It is important to mention that since the parties involved were in different departments, a high degree of communication needed to be in place since they all may have had different mandates and goals.

Large Latin American bank

This multinational bank organized its support structure the same way it dealt with new technologies, much like the previous case. The difference was that upper management did not provide full support to the team, as noted by a lack of responsiveness during some troubleshooting woes. Ultimately, the issues were resolved but it was made clear that a tighter commitment from management was needed for the project to gain credibility.

In this case, the support group was composed of capable architects, a system administrator and a part-time project manager. They were all part of the same support organization, and their responsibility was not only WebSphere but also other types of middleware products. They were under tremendous pressure to deliver a solution, while being understaffed and budget-challenged.

Lesson: *If upper management is not one hundred percent behind the support effort, it subtracts credibility and needed influence within the organization.* The author Ed Yourdon once said "All projects are behind schedule – it is just a question of when you discover that fact and how the project team will respond to the discovery." While this might be a bit extreme, it does bring out the idea that projects within themselves provide enough material to remain busy. The effort of the people involved should be focused on solving those inherent issues, as opposed to worrying about more fundamental items such as whether or not management is fully backing the effort.

Large financial company

At this company, there is no central group that provides support to the application groups, delivers standards, and manages the dissemination of WebSphere knowledge. This has led to application group architects with strong personalities charting the direction and choices for applications, without much input from anyone else. Although those individuals had some experience in the field, several situations occurred in which they disagreed with other architects, and there was no procedure in place to help in resolving the issues. Consultants from IBM were employed from time to time to provide onsite, level two type of support. The systems administrators also sometimes got into conflicts with the development groups, as they wanted a specific version deployed, while the other groups said it was not possible.

Lesson: *A centralized decision-making body, with the authority to chart direction on WebSphere matters, is needed in order to avoid unnecessary arguments.* Although the company had all of the necessary “pieces”, they tended not to fit well together. In this environment it is very easy to put individual agendas ahead of the company’s best interests.

Solution

As shown by these cases, customers have implemented their WebSphere support groups in many different ways, and some worked better than others. In the following sections we describe a support group that is composed of the best elements from the previously described experiences.

The name WebSphere Engineering Group is used throughout this document. “Engineering” denotes the application of science for practical purposes, thus its use emphasizes the need for a more disciplined approach to developing and supporting software. The name also provides a clear separation from other internal groups, like Architecture.

Details

One of the key purposes of this support group is to help lift the confusion that sometimes occurs when trying to implement new technology by providing sound, tested advice and support. In order to maximize this task, the following four areas that the group should address have been identified.

Management

Non-technical issues influence the outcome of a software support effort as much as the technical ones. The WebSphere Engineering Group should contain a project management skill set to address non-technical issues as they arise. This is very important, as those items could potentially take a long time to resolve. Lines of business are usually run by function, whereas IT personnel tend to be driven by schedule, so the reconciliation of these two approaches should be encouraged by this initiative.

This is essential in order to help deliver solutions that are on time, within budget, and that contain not only the right features, but also the right level of quality. Without proper management, a support group would find it hard to keep focused on the greater picture when engaged in helping WebSphere development groups.

Planning

This initiative should provide knowledge of future WebSphere features, possible hurdles, and direction in general. It should promote understanding of the company's position on relevant technologies, and advise accordingly. It would also provide advice regarding migration efforts, platform selection rationale, tools, and educational needs. This initiative would serve as the buffer between the internal WebSphere community and vendors, understanding how to satisfy pressing and future business needs with WebSphere offerings.

Processes

This initiative has the mandate of delivering relevant processes and standards that would streamline and optimize the administration and development of WebSphere-based applications. It would also be the knowledge management gatekeeper, providing for an environment conducive to intellectual capital generation. It would provide for the reuse strategy, identifying what and how WebSphere artifacts should be reused.

Services

This initiative will provide centralized help for WebSphere development groups, and allow for early enforcement of proven design techniques, standards and architectures (that is, artifact reuse, documentation, pattern adoptions, and so on), serving as the first line of support and working with developers to resolve difficult, key troubleshooting incidents. This group would be charged with documenting the consulting tasks that it launches, thus helping to implement the knowledge management effort laid out by the process initiative.

Figure 1 1 illustrates how these four areas interact with each other. The group should be a very dynamic, fully interactive element.

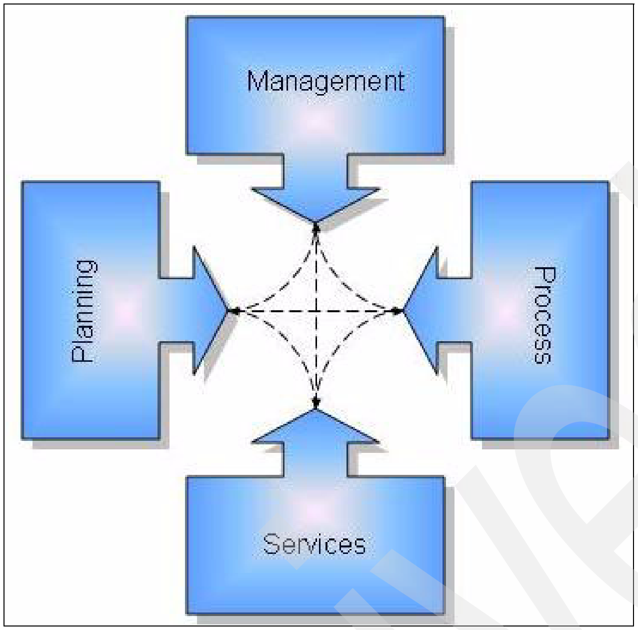

Figure 1 Support group interactions

Each part is aware of the work of the others and thus, they provide a synchronized support effort. This is particularly key for the services area, because it is the most visible part of the group. The services area interacts with the development groups most often and in a more in-depth way. The message that the services area delivers to development groups needs to be in full compliance with the work executed by the other areas.

These elements, working together, provide a strong, thorough approach to support delivery. They allow the company's other resources to focus on what they do best—instead of trying to provide an uncoordinated effort that (although with possible good intentions) would not allow for maximum utilization of resources. For example, z/OS system administrators today may be involved directly with development groups if there is no one else to interface with them. The system administration group may also be involved in multiple iterations of testing in which they should not normally have to be involved. Some of these issues should have been resolved before the application was delivered through them.

In the new scenario, the WebSphere Engineering Group would be there to help the application groups, contacting the system administrators only when really needed.

Roles

We previously discussed the key initiatives that the support group should address. In this section, we concentrate on the specific skills that the engineering group members should possess.

Program manager

The program manager should have proven consulting experience in managing complex, multidimensional programs. Experience in managing relationships at various levels, including the senior executive level, will be expected. A successful track record of managing contractors and vendors would be desirable, together with managing individuals at different locations. Depending on the size and number of projects being supported, this position could be half-time at the beginning, although a full-time position is encouraged.

Process engineer

The process engineer's main role is to provide processes and standards to better optimize the WebSphere experience at the company. These issues are not tilted towards optimizing one specific development group, but rather to improving the environment for all applications.

This position requires an understanding of the current processes and standards with regard to the software life cycle adopted by the company. Having a solid understanding of software development methodologies (Rational Unified Process®, Extreme programming, and so on) together with common reuse strategies is strongly recommended. A background in quality assurance and testing would be desirable, and knowledge of J2EE architecture and legacy systems is a necessity.

Service engineer

The service engineer would provide mentoring and hands-on support to the development groups. Deep experience in translating business requirements into technical solutions would be required. This senior level consultant should have successfully completed many large scale development projects. Having solid writing and verbal communication skills, together with a clear understanding of the "soft" skills utilized in a consultant context, would be desirable. Expert knowledge of Java and J2EE, together with experience with legacy systems and integration, should be key prerequisites.

Planning engineer

The planning engineer would help in the migration planning of the different WebSphere runtimes. This individual should understand the future plans for WebSphere, and provide a pragmatic view of the rest of the industry. A keen interest in and knowledge of new trends in the industry should be encouraged, and experience in a role establishing strategic directions would be preferable. Advanced knowledge of J2EE and legacy integration would be a requirement, and having solid communication and presentation skills would be key.

Note that the production and test WebSphere systems are maintained by the regular infrastructure groups; there is no role of “system administrator” within the engineering group. However, the planning and service engineers would work very closely with specific system administrators when solving specific issues, as needed.

Figure 2 on page 10 depicts the proposed solution and how it will fit within the organization. Note that the traditional infrastructure and operations roles remain unchanged. What is different is that some of the tasks that were being fulfilled by the infrastructure groups (that is, advising development groups and researching new WebSphere releases) have been moved to the WebSphere Engineering Support group.

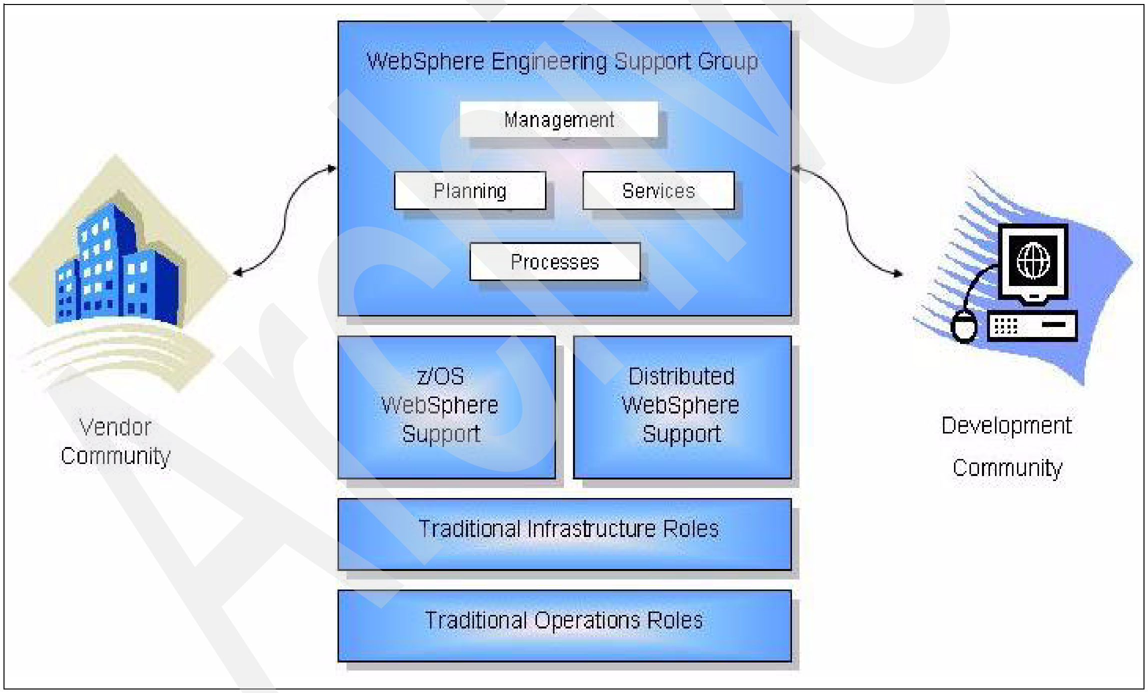
WebSphere Engineering Support Group
Management
Planning
Services
Processes
z/OS WebSphere Support
Distributed WebSphere Support
Traditional Infrastructure Roles
Traditional Operations Roles
Vendor Community
Development Community

Figure 2 Support group overview

The production deployment of WebSphere applications, like any other production application, is still managed by the operations group. The application development groups would still need to interface with the operations group, since they have to provide installation scripts and other artifacts the operations group might need.

This process could be facilitated by the service engineer assigned to that specific development group. The WebSphere vendor community (IBM, third-party vendors) would also be managed through the group. In situations where the product falls within the jurisdiction of another internal group, like WebSphere Application Developer (WSAD), close interaction should occur between the WebSphere support group and the internal entity.

Notice that the engineering group serves as a proxy between the internal development community and the vendor community. This helps in the management of standards and dissemination of pertinent information.

Staffing

The minimum number of people needed for the support group would be one for each of the roles described in “Roles” on page 9. As groups needing help to start developing become more common, more service engineers should be added. A rule of thumb is to have about five full-time man hours.

This could consist of one project manager dedicated to the project part time, a process engineer also working part time, plus three service engineers and one planner engineer on full time. This would represent five full-time man hours (even though physically there are six individuals). The actual size of the staffing would depend on the present and forecast number of groups to support.

It should be noted that the process engineer would not be crafting processes in isolation, but rather working in tandem with the group which manages the software development life cycle at the company. This method helps to ensure that the planned methodology is followed, with the process engineer making the needed modifications, if necessary, from a WebSphere context.

The role of the process engineer is not ongoing—it is mandatory when the group is starting, but becomes optional once the group reaches maturity. Processes are not things that are developed all the time. Once they are in place and are proven effective, occasional reviews should occur. Thus, this position should be conceptualized as having an “as needed” nature.

Physical representation

Up to this point, we have discussed the need to establish a WebSphere support element within the enterprise, the type of expertise that is necessary, and the number of staff to optimally fulfill the need.

The next step is also the most political: where is the support group to be located within the company' structure? This can be a very sensitive item indeed as there are two important contexts to consider: the nature of the location of the group (that is, would a standalone department be better than a virtual team approach), and the nature of the staffing of the group (that is, would it be better to utilize in–house resources as opposed to contractors or even outsourcing).

The answers to these questions depend on the type of culture, situation, support, and leadership currently in place. An independent entity is preferred over a virtual team ("virtual" in this case means different organizations within the company).

One key element in determining the success or failure of a support group approach is the communication level achieved among the different parts involved. It is much more difficult for people to give the necessary amount of work to the support area if they have other responsibilities to fulfill in their respective departments. It is also much easier to create stronger bonds and provide a more cohesive, synchronized approach towards providing solutions if the members of the support engineering group are located within the same department. A virtual team could be a valid choice in situations in which it is not possible to create a new administrative entity.

The goal is to have a set of in-house, full-time personnel comprising the support team. If that is not possible, interim external consultants can be used to bolster in-house staff technical skills The extent of the consultant engagement should be proportional to the level of in-house expertise available (a typical range is from several weeks to gain a very specific single skill, to about a year for a thorough transformation of the way in which issues are approached and solved). A detailed expertise transfer agenda, with specific metrics, should be put in place before external consultants are called in.

IBM is uniquely qualified to help the support group at this stage, through individual consulting or through the WebSphere Center of Excellence offering.

Outsourcing should only be considered if a company has the necessary expertise in coding and design to truly be able to manage the relationship. Companies should avoid becoming too dependent on the outsourcing company by having the expertise in-house to analyze deliverables and determine met expectations.

Other mandates

Reuse, knowledge management, and quality assurance are three important challenges that are spread throughout the proposed group. The responsibility for achieving each one of them is shared by all the group members.

It is important to note that the responsibility to implement these mandates at the enterprise level is not the responsibility of the support group. If there are processes in place to implement them, the support group should mold their own initiatives in this area to fit into what the company's targets are.

However, if the company does not have activities in place to deliver these mandates, the support group should not ignore them but instead, implement them locally. When the business is ready to implement them company-wide, the support group should be involved in their crafting, as they would have valuable hands-on experience on the subject. These mandates as sometimes referred as being *orthogonal* to the mode direct responsibilities.

Reuse

A corporate culture that cherishes and promotes reuse is only achieved through hard and well-planned work. At the very foundation of a reuse strategy is *trust*. Without it, developers will tend to try to develop functionality from scratch whenever possible and project managers will feel that the risk involved with artifacts flagged for reuse is too great to justify their use.

It is a common mistake to let developers be fully in charge of reuse. This strategy tends to produce unsatisfactory results (even though some code or class level reuse might be achieved). This is because a developer's focus is on deliverables for a specific project, while reuse is a strategy not tied to any specific project but to the good of the enterprise as a whole.

A successful reuse strategy starts at the business layer, not at the technical layer. It is here that business analysts and individuals knowledgeable of the internals of the company processes and organization identify the different areas of possible reuse, which are usually referred as *domains*.

Once an enterprise is divided into domains, specific opportunities for reuse are further subdivided by defining the different functional processes that allow for work to be accomplished. Once these processes are identified, the various activities and tasks that compose those processes are documented. This will serve as a starting point to identify interdependencies and possible repeatable actions.

For example, in both the Accounting and Personal Finance departments, there is a requirement for logging certain types of key activities. This could be the beginning of an "auditing" reusable component that could be shared between departments.

After domains (with their processes, activities and tasks) are identified, efforts can move to the technical level. After reusable entities are produced, they have to be managed, as addressed in the following section.

Knowledge management

Knowledge management deals with how to best take advantage of present expertise in order to optimize the achievement of business objectives. The chosen reuse elements need to be organized and advertised in such a fashion that they are easily obtainable and traceable.

Being *obtainable* is obvious, as it is impossible to reuse something that you do not know exists. Being *traceable* helps to track who has and currently is using the given component (for example, to alert users if a new version of the component becomes available, or if it becomes obsolete).

It is key to identify early on who is going to be in charge of managing the effort and what technologies the company has available for artifact management.

Knowledge management extends much further than just component management; it deals with managing all kinds of intellectual capital. Another type of capital is documentation.

- The service engineers should ensure that the knowledge gained throughout the different engagements with the development groups is documented. The process engineers should make sure they document their processes and methodologies, together with any other guidelines that may benefit the software life cycle in a WebSphere context.
- The planning engineers need to document their views of competing development approaches and issue a statement (one example is JDBC versus SQLJ). Updates of a master document on WebSphere futures and a recommended direction to be followed by the company should be done per major update of the WebSphere runtime or J2EE programming model.
- The program manager needs to ensure this kind of documentation and others are actively developed, especially after each assignment. This should be complemented by frequent (monthly would be a good time frame) and comprehensive report analysis, specifying trends and present and future issues.
- A weekly high level status report of the program is also recommended.

All of this documentation needs to be organized in a manner that results in an easy-to-find and easy-to-share repository of WebSphere knowledge.

Quality assurance

It is of little use to the company if there are efficient processes for reusing components, and excellent ways for those components and their documentation to be made available to the development community, if those components to not meet a desirable level of quality.

If products are of poor quality, reuse will not be achieved at the level necessary for a successful return on investment. Quality assurance is much more than just quality testing; it entails the ongoing understanding that quality permeates *all* facets of software development, not just testing (which is sometimes referred to erroneously as “quality assurance”).

Quality assurance has enforcement and planning aspects, as well. This implies that it has to be defined in order for it to be able to be enforced, and that unless it is explicitly planned, its effectiveness will be diminished.

There are two predominant views regarding quality management:

- The first view is that since quality is the job of everyone, there is no need for it to be treated separately. Each individual involved should be responsible for producing quality work. This will produce an enhanced final deliverable.
- The second view is a bit more skeptical: quality work is the responsibility of everyone, but since quality is an element that is hard to measure (as opposed to costs, features and time), it is usually the first element to be ignored if a schedule slips.

  And since there is no one party responsible, there is no single point of control and enforcement. Also since no one is in charge of managing quality from an end-to-end perspective, it is easy to lose track of what is best on a larger scale.

In the second view, the term “quality manager” is often used. This individual (who probably also has other responsibilities) would help craft a set of quality control measures and metrics in order to track and understand more efficiently the state of quality at the different stages of the project life cycle—and execute all of this with the vision of what is needed at the enterprise level. This individual would serve as a liaison to the other quality assurance efforts throughout the company.

My preference is for the second approach. The program manager should be the person keeping track of the level of quality that the support group is exhorting, while the process engineer should deliver recommendations on how to improve quality throughout the development and deployment of WebSphere applications.

Conclusion

In this Redpaper we analyzed the reasons why the addition of an internal, specific support group is encouraged for companies wanting to create a mature, scalable, enterprise-level WebSphere environment. The more, and better, support a company provides to its development groups, the more confidence and success they will have with WebSphere itself. We also explored the details of the proposed group, and highlighted areas that should be taken into account when crafting the group's agenda.

In summary, the five key recommendations are:

- Create a WebSphere Support Engineering Team.
- Establish different support roles, with focused responsibility.
- Obtain and maintain strong support from senior management.
- Keep in mind orthogonal mandates.
- Review lessons learned often.

The role of the WebSphere Support Engineering Team is pivotal. It allows a corporation to be able to minimize risks by effectively developing, governing, and enforcing the processes and artifacts that make solving business problems, in a WebSphere context, easier.

Proper support for the WebSphere engineering group at the senior executive level cannot be emphasized enough. This support needs to be in place in order for the program to be fully effective. If the group does not have the ability to influence and enforce its deliverables, then its usefulness is greatly minimized.

The group's shape and size varies greatly from customer to customer, but the skills needed for the basic core team are widely applicable. The suggested staffing size rule of thumb given in this Redpaper is a good starting point.

The mandates specified in "Other mandates" on page 12 (reuse, knowledge management, quality assurance) are key qualities present in mature, successful endeavors. They should not be considered merely optional, but as part of the core set of elements that the support group encourages.

It is important to recognize where others have faltered, so you can avoid repeating the same mistakes. It is advisable to re-read the "lesson" sections from time to time to ensure focus is still in the right direction. Documenting some of your own "lessons learned" would be an excellent way to keep track of possible items of concern specific to your own environment.

New technological changes like WebSphere come around every ten years or more, and using a WebSphere engineering group to improve the support process offers an opportunity to "get it right the first time". If corners are cut, it could lead to more expensive propositions later and certainly, a more risk-prone endeavor all along.

The author of this Redpaper

Jorge Diaz has worked as a senior consultant, architect, and engineer. For the past decade he has focused on the areas of distributed, object-oriented software architecture and development. At IBM, he provides consulting for the WebSphere® family of products throughout the Americas and Europe.

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:
IBM Director of Licensing, IBM Corporation, North Castle Drive Armonk, NY 10504-1785 U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:
This information contains sample application programs in source language, which illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.

Trademarks

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

- IBM®
- Rational Unified Process®
- Redbooks (logo) ™
- WebSphere®
- z/OS®
- zSeries®

The following terms are trademarks of other companies:

Intel, Intel Inside (logos), MMX, and Pentium are trademarks of Intel Corporation in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

SET, SET Secure Electronic Transaction, and the SET Logo are trademarks owned by SET Secure Electronic Transaction LLC.

Other company, product, and service names may be trademarks or service marks of others.