WebSphere Engineering

A Practical Guide for WebSphere Support Managers and Senior Consultants

Ying Ding
Introduction

My experience in applying WebSphere® Application Server technologies has been a journey of making many mistakes and learning from them. This book, in a sense, is both a landmark of my journey and a systematic summary of a lifetime of learning opportunities.

WebSphere® Engineering: A Journey of Learning

In 1999, I started to work on WebSphere Application Server 3.0 as an IBM technical consultant. After braving a month-long intensive boot camp at a training facility covered by deep snow outside of Washington D.C., I was deployed to several large Texas energy companies. These engagements were great learning experiences. I was not only submerged in WebSphere technologies, but I also had a tremendous professional education that focused on customer satisfaction and depended on a tight team to deliver quality IT products and services.

In 2001, I joined a large financial services company and soon became a senior technical manager. Over the next five years, I stayed in the field of WebSphere Application Server engineering. I was close to the teams, the technologies, the systems, and the engineering processes. As a frontline practitioner, many years of technical and management engagements have been critical to building my expertise of WebSphere technologies. It allowed me to develop an incrementally well-rounded insight into the key aspects of WebSphere engineering. My work in the field exposed me to the full life cycle of WebSphere Application Server deployment and helped me increase my knowledge of WebSphere engineering.

After several years of achieving high WebSphere system stability for large and capable WebSphere Application Server systems, my first series of articles on WebSphere engineering were published in 2005. In my pursuit of 100 percent WebSphere system stability, I realized that the WebSphere technologies were stable, mature, and reliable; the technical skills of WebSphere engineers and consultants were good. However, a troubling and provoking question remained:
Why were there still so many production instabilities and so many stressful and demoralizing production fires in the industry? I struggled to understand this issue. Eventually, it dawned on me that WebSphere system instability was primarily a result of quality issues in product and service delivery. On further examination, I realized that these quality issues are directly related to the lack of rigorous system standards, consistent engineering processes, problems in hiring and training, and organization dynamics and stability issues. The challenges in WebSphere engineering practice and system stability had their root in the technical management of WebSphere Application Server engineering or, rather, the lack of it. This basic observation motivated me to document my thoughts and share them through publication. With continuous research, the concept of WebSphere engineering has become more defined and has led to this book. This book’s intent is to identify where improvements in WebSphere engineering practice can be made and how to make these improvements. The chapters present various aspects of WebSphere engineering.

This Book’s Organization

This book is a systematic introduction to WebSphere engineering. It provides an inclusive body of knowledge, best practices, and experienced insight as to where to make improvements. Its objective is to help WebSphere engineering support managers, senior WebSphere engineers, and consultants. This book covers the critical aspects of WebSphere Application Server infrastructure engineering work, from engagement to production operations.

This book is not a theoretical discussion of middleware engineering. Instead, it is based on industry experiences and, sometimes, the author’s and technical reviewers’ painful lessons of WebSphere Application Server engineering practices.

Many chapters include a section that discusses lessons learned or critical issues pertaining to the chapter’s main topics. This section may provide a systematic study of WebSphere engineering or useful references for on-the-spot solutions. Chapter 4 through Chapter 10 provides WebSphere engineering operations. These operations help ensure the quality delivery of WebSphere products and services.

Chapter 1, “Organization Models and Choices,” discusses organization options and their pros and cons. These organization models provide references that help with organization design and redesign to achieve the optimal organization structure for your WebSphere support teams. This chapter defines WebSphere engineering tasks or operations. You must understand the classification of these operations because they are used in numerous important topics, such as engagement. This chapter explores the relationship between the organization model, organization stability, and WebSphere system stability.

Chapter 2, “Building a World-Class WebSphere Team Through Hiring and Training,” focuses on technical skills. It discusses technical skills in four areas:

- A balanced approach to skills that include technical skills, technical project-management capabilities, and communication and teamwork experience
• Team building through structured and experienced hiring practices
• Focused technical training with different delivery models and maturity levels, and effective training methodologies and resources
• Technical training to grow the team’s skills and serve as a morale booster for the team and as a relationship builder between the manager and the team

Chapter 3, “WebSphere Operations Framework,” explores the core concept of WebSphere engineering and presents the WebSphere Operations Framework. Carefully read this chapter to understand WebSphere engineering and the WebSphere Operations Framework. This chapter expresses the intent to bring WebSphere Application Server work closer to a discipline of engineering that is characterized by tight standards and consistent processes.

Chapter 4, “Engagement Challenges,” helps you secure adequate financial resources for your WebSphere team, both at the beginning and in the middle of a large project. This chapter discusses many systemic IT issues that make good engagement difficult to achieve. This chapter presents a service assignment and tracking system to quantitatively measure WebSphere services by appropriately using WebSphere engineering operation classification. It recommends a three-phased engagement process to manage large and challenging projects and an initial sizing and consulting engagement operation. This chapter shares experiences in managing dynamic and large projects, which cannot be well defined at the beginning of a project.

Chapter 5, “Server Build,” discusses system-build issues and recommendations to deal with them. This chapter includes three operations: build planning, system design, and system build and configuration. This chapter also shows you how to validate your system build through testing. The descriptions of these operations conform to the WebSphere Operations Framework.

Chapter 6, “Functional and Integration Testing Environment Support,” discusses major testing environment WebSphere support issues. This chapter explores the options to support testing environments, such as whether Java™ consulting needs to be provided or whether to use WebSphere Virtual Enterprise or WebSphere Application Server network deployment. This chapter also covers the Functional and Integration Testing WebSphere Support operation and resource and workload considerations.

Chapter 7, “Stress-Testing Environment Support,” introduces the stress-testing WebSphere environment. This chapter uses the WebSphere Operations Framework to depict two WebSphere engineering operations: WebSphere Application Server configuration change support operation and WebSphere Application Server stress-testing support operation. This chapter emphasizes the importance of stress testing and explains the common problems resulting from insufficient stress testing. It discusses the major challenges of the stress-test environment for the WebSphere team. This chapter also explores the relationship between the production environment and stress-testing environment. Finally, this chapter sheds some light on the collaboration for stress-testing environment work and provides insight into the direction of the WebSphere stress-testing environment engineering support.
Chapter 8, “Production Environment Support,” starts with lessons learned in maintaining WebSphere production system stability. These are practical insights into several areas of production support. Some experiences are counterintuitive, but nonetheless important. This chapter then offers important best-practice recommendations, such as empowering the production support team by sharing knowledge, system privilege, and tools. Finally, it describes the WebSphere Application Release Support Operation and the WebSphere Data Center Switch Support Operation.

Chapter 9, “Managing a Production Emergency,” includes topics such as evaluating the severity of a problem and problem reporting, managing the WebSphere team during a production emergency, mitigating customer experience, and the all-important task of effective communication during high-impact and high-profile production problems. It also discusses how to build and strengthen your work relationship with IBM, as well as the formal process for further correcting, resolving, and preventing the problem from recurring (post-problem resolution). When you experience a production problem, read this chapter to obtain practical help to better manage your emergency.

Chapter 10, “WebSphere Application Server System Upgrade and Product Maintenance Management,” first covers important system-upgrade challenges, such as obtaining approval, scheduling, adhering to standards, and not upgrading just for “upgrade’s sake.” This chapter discusses WebSphere Application Server planning engineering, process engineering, and build standards. Upgrade strategy communication is a major topic, especially from the aspect of gaining your business partner’s support. This chapter also introduces the WebSphere Application Server Product Maintenance Operation.

Chapter 11, “Critical Work Relationships,” covers the work relationship with the enterprise architecture team, the testing organization, application development and production support teams, and the capacity planning team. It also discusses nontraditional system resources, such as the web container thread and database connection pool.

Chapter 12, “Managing the Stability of Large Enterprise WebSphere Systems,” stresses the need to intensely focus on and proactively manage the stability of large enterprise WebSphere systems. It provides insight and recommendations on achieving a balance between addressing immediate stability issues and long-term stability strategies. This chapter introduces ways to fundamentally improve large middleware system stability by forming the right financial incentive plans and organization structures. In addition, this chapter explores strategies to improve system stability for large enterprise WebSphere systems in interconnected change management.

Chapter 13, “WebSphere Engineering Going Forward,” examines the future of WebSphere engineering. This chapter explores three new areas:

- WebSphere technologies that provide end-to-end IT infrastructure for SOA
- WebSphere Virtual Enterprise
- Service Science Management Engineering
Chapter 13 briefly introduces new WebSphere products and technologies, as well as the basic concept of Service Science Management Engineering. However, the focus is on technical skills, organization models, and engineering processes. This chapter purposely does not provide any definitive conclusions about WebSphere engineering’s future, but asks intelligent questions that may lead to productive and systematic discussions of WebSphere engineering going forward.

How to Use This Book

This book can be read as a holistic study of WebSphere engineering or it can be a reference for on-the-spot solutions. For example, when you are in a production emergency, you may find practical help by reading Chapter 9. In addition, in Chapter 1, you may see a WebSphere organization model that makes sense to you when you are working through an organization redesign.

Although many chapters can be read independently, it helps if you read at least Chapter 1 through Chapter 3. These chapters familiarize you with the WebSphere engineering terminology and the basic ideas of WebSphere engineering organization, classification of operations, the contents of WebSphere engineering, and the WebSphere Operations Framework structure. As a result, it will become easier to read the rest of this book, even if you only read certain chapters.

I hope that this book provides you with helpful insights, recommendations, and references. If this book can help you better manage one problem in your WebSphere engineering practice, then my time writing this book is well spent.
As a leader of WebSphere Application Server engineering, the first challenging task that you have is to either build a new WebSphere organization or to streamline an existing one. Then, you need to choose a suitable organization model.

This is a tough job because the stakes are high. An unsuitable organization model can lead to an inferior WebSphere organization. Depending on the model that you choose, the WebSphere organization will either be well organized or intrinsically flawed. For example, a WebSphere organization with many unnecessary layers of escalation for production support won’t be able to address production problems in a timely manner. In addition, unmerited division of engineering tasks, such as the separation of the project interface and system build into different teams, can cause serious relationship issues within the WebSphere organization because of resource contention and team priority differences.

The choice of organization model affects the stability and availability of your enterprise WebSphere systems. In addition, you may have to make your model choice and implement it while mergers, acquisitions, and other business changes take place.

This chapter explores different models of a WebSphere organization, including the following:

1. A WebSphere organization is a shorter way of saying a WebSphere Application Server engineering organization. It is responsible for the design, build, and operation of the critical WebSphere infrastructure of a large company.

2. A WebSphere system is the short form of WebSphere Application Server system. This term emphasizes the WebSphere Application Server system software installed and configured for a specific target JEE application or application suite. “WebSphere system” as a technical term or concept does not include the JEE application or the JEE application suite that executes in the containers provided by the WebSphere system.
• Dedicated WebSphere organization
• Line of business (LOB)\(^3\)-based support model
• WebSphere organization with separate engineering function
• Global WebSphere workforce
• WebSphere support of multiple levels
• WebSphere support for large projects with multigenerational plans
• WebSphere Center of Excellence

**Dedicated WebSphere Application Server Engineering Support Organization**

Before getting into a detailed discussion of a dedicated WebSphere organization, we need to define some terms: product-based support model and dedicated WebSphere support organization.

A *product-based support model* refers to an organization structure used in an IT infrastructure engineering organization. This type of structure is responsible for the full life cycle of the product engineering for one given set of products and related technologies. For example, you have a product-based technical support team that is responsible for the engagement, design, build, operations, support, and decommission processes for WebSphere technologies such as WebSphere Application Server, WebSphere Process Server, WebSphere Enterprise Service Bus, and WebSphere Portal. This technical team would not work on databases, operating systems (OSs), messaging technologies, and so on.

The use of the word “dedicated” in *dedicated WebSphere support organization* has two meanings. First, the expression “dedicated” indicates a technical team that is specialized in WebSphere Application Server infrastructure engineering. In other words, WebSphere Application Server and related technologies need to be this team’s only concern. For example, building and supporting JBOSS and WebLogic need not be the concerns of a dedicated WebSphere team.\(^4\) In this perspective, a dedicated technical team belongs to the product-based support model.

The second dimension of the term “dedicated” refers to a WebSphere Application Server engineering team or WebSphere team that works only to support its assigned LOB. A key organizational difference for a dedicated WebSphere team is the reporting structure. These WebSphere teams may belong to one large system-wide WebSphere organization, or these teams may have no horizontal organization connections. They report respectively to different IT divisions working for different LOBs.

---

3. *Line of business (LOB)* frequently refers to a highly connected group of products and services and business organizations that provide such products and services. For example, the consumer banking organization of a large international bank would be such a LOB. A large LOB may have many large business divisions, and each such business division may have a dedicated WebSphere engineering support team. A consumer online banking division of a large bank is a good example of large business division under a LOB.

4. *WebSphere team* is short for WebSphere Application Server engineering support team. The WebSphere team belongs to a WebSphere organization.
Separation of Teams and Classification of Tasks

A WebSphere organization must be divided into planning, process, and service teams. The planning team provides product strategy; the process team works on standards and engineering processes; and the service team delivers WebSphere products and services.5

The WebSphere engineering tasks are classified into three categories that have detailed operations. These categories are as follows:

WebSphere Planning Engineering (Plans and Strategies)
- Evaluate target legacy systems and form conversion strategies and plans.
- Evaluate existing WebSphere systems and devise migration strategies, plans, and roadmaps.
- Evaluate industry trends and emerging technologies and form introduction strategies and plans for the approved emerging technologies.
- Evaluate Java Enterprise Edition (JEE) standards development and give advice on migration strategies and plans.

WebSphere Process Engineering (Standards and Processes)
- Server build planning
- Server build
- Security
- Design naming convention
- Documentation
- System certification and validation
- WebSphere design and configuration
- WebSphere application deployment
- Integration methodology
- Development methodology
- Scripting and automation
- WebSphere best practices

ENGINEERING TASKS ARE DIFFERENT BETWEEN TEAMS

Many tasks for WebSphere process engineering and WebSphere service engineering may have the same names (for example, server build). However, the “server build” operation is a different task to different WebSphere teams. A team focusing on WebSphere process engineering is concerned with the WebSphere server build standards, the server build process, and process automation. A WebSphere service engineering team applies the WebSphere server build process and automation programs to build, configure, and deliver the WebSphere servers.

WebSphere service engineering (service delivery) has many similar tasks as WebSphere process engineering, but different contents. Here are the engineering tasks specific to service engineering:

- Security enabling
- On-call assistance
- Implement naming convention
- Troubleshooting and problem resolution
- Performance tuning and testing
- Development support
- WebSphere consulting
- JEE consulting

Categorizing WebSphere Application Server engineering functions helps conceive and design the structures of a WebSphere organization. The enumeration of engineering tasks makes it easier to think through the convergence of WebSphere Application Server engineering support life cycles, WebSphere Application Server environments and their components, and the product and service delivery.

This category of engineering function and the classification of engineering tasks are necessary elements that comprise an engineering framework. They make it possible to systematically deal with WebSphere Application Server engineering tasks in concrete terms. Chapter 3, “WebSphere Operations Framework,” introduces the WebSphere engineering framework.

Although planning engineering and process engineering functions have different engineering tasks, they do have one similarity. These WebSphere Application Server engineering functions have no direct contacts with the customers. WebSphere teams serving these functions do not deliver WebSphere products and services directly to business partners. This is why frequently one WebSphere team takes on the responsibilities of both WebSphere planning engineering and WebSphere process engineering. However, from time to time, it is necessary to use senior planning and processing engineers to assist with high-impact production problems or difficult technical problems, working directly with the customers.
Dedicated WebSphere Organization of a Product-Based Support Model

A large, dedicated WebSphere organization that conforms to a product-based support model has many benefits. This organization is dedicated to WebSphere Application Server and related technologies. Figure 1.1 describes this organization model. A high level of communication helps achieve the standardization of systems and consistent engineering practices. A central leadership can effectively enforce the engineering processes and procedures across the WebSphere organization. A common document repository, a focused Web site, and a consistent documentation process contribute to successful knowledge management practice.

A service-oriented and project-focused approach can help with potential organization inflexibility associated with adopting a product-based support model. A project-oriented performance management needs to be considered as part of the complete solution of a dedicated WebSphere support organization.

![WebSphere Organization](image)

**Figure 1.1** Dedicated WebSphere support organization built on a product-based support model

For a mature IT organization with an effective engagement process and a transparent IT infrastructure engineering cost model, use a dedicated WebSphere Application Server engineering support organization built on a product-based support model. This organization model works particularly well if your company has mature project management practice and large interconnected information systems.

However, a product-based support model may not work well if your engagement process is still evolving. A large technical organization dedicated to WebSphere Application Server and related technology may not be flexible enough to adapt to the fast-changing business needs of many projects. For example, it may be a challenge to quickly redeploy engineers to respond to the
sudden surge of resource needs for a highly dynamic project. In addition, this support model works better if you have a sophisticated and transparent cost model. A large centralized WebSphere organization adds another layer to an already opaque financial metric. It is more challenging to depict a clear picture to your customer of WebSphere Application Server engineering costs.

Therefore, other important organization models deserve proper consideration (for example, the LOB-based support model).

**LOB-Based Support Model**

There are two different ways to implement a LOB-based WebSphere Application Server engineering support model. The first way is to build a technical team that supports multiple technologies and products for one LOB or a large business division. For example, you can build an infrastructure engineering team that supports WebSphere Application Server and other middleware products, the OSs, and the database technologies. Let’s call this model the infrastructure engineering solution team, or to be more concise, the solution support team, because it is the convergence of many different technologies to engineer an IT infrastructure solution for the customer.

The other choice is to build separate infrastructure engineering teams that are dedicated to WebSphere technologies, but belong to different technical organizations that support one LOB or a large business division. For example, you may have one WebSphere team that supports an IT division working for the customer relation management (CRM) department. You may have another WebSphere team that is dedicated to the IT group for your sales and service division. These two WebSphere teams report to different IT organizations in your company. They do not belong to a company-wide WebSphere organization. There are no organization connections between these WebSphere teams.

These two organization models share some similar advantages and disadvantages. However, because there are enough differences, they are discussed separately with the similarities compared from time to time.

**Infrastructure Engineering Solution Team**

The major advantages of this organization model are flexibility, streamlined support, and ease of doing business for business partners and customers. The major disadvantages are that this form of organization is less developed and it needs work and time to grow and mature.

**Advantages of a Support Team of Multiple Products**

The relatively smaller infrastructure engineering organization is agile and flexible. In comparison, because of its size and complexity, a large IT infrastructure engineering organization may not be able to plan and implement changes as nimbly. A relatively smaller infrastructure engineering solution team may be able to adapt faster to changing business drivers; therefore, it may be able to do a better job of serving the business. The same goes for a small, separate WebSphere team dedicated to a LOB. Figure 1.2 depicts this support model.
Better coordination is also an advantage of the infrastructure engineering solution team. For a product-based support model, you need special teams to coordinate the work of many product-based technical teams. For example, you need dedicated change coordination and environment coordination teams to ensure that the technical teams work together seamlessly. This is especially important during critical system changes, when a good transition between technical teams is critical.

**Environment Change Coordination Team**

For a large technical environment in which many technical teams work together to perform complex changes, the environment change coordination team (also called the technical environment coordination team) leads the planning, scheduling, and coordinating of the changes through development and testing pipelines. This team plays a critical function that is indispensable for high quality and efficient change execution. For example, this team can help with the important communication work between technical teams. After an OS upgrade, this team can help certify the OS change and then inform the WebSphere team that it can commence its change. Without this team, timely communication and coordination may not consistently happen, which, as a result, may cause delays, quality issues, and even system production outages.

Good coordination is essential in managing large and complex technical environment. When you choose to build dedicated technical teams for each core technology, you want to establish a coordination function to help the technical teams work together. For any of these technical teams, doing a good job as an individual team is not good enough.
For example, during a major system upgrade that takes a long time, if the UNIX® team fails to inform the WebSphere team to begin WebSphere configuration changes as soon as it finishes an AIX® upgrade, the WebSphere team may not have enough time to perform a complex WebSphere system reconfiguration as planned.

An infrastructure engineering solution team can better deal with work coordination. For this support model, the system engineers for different core infrastructure technologies belong to the same team, and the coordination between them becomes substantially easier. In addition, for such a team it is unavoidable that one system engineer supports multiple core technologies. For example, the WebSphere engineer6 may also serve as the UNIX system administrator. Then there won’t be coordination difficulties.

In addition to better coordination, a single point of contact is highly desirable for good work relationships with your business partners and customers. It makes working with your team easier. Remember, from their perspective, the situation when multiple contacts to the infrastructure engineering organization have to be made for technical assistance must never arise.

For example, your application development team may feel burdened if it has to contact the WebSphere team, the database team, the operating system team, the load balancer team, the DMZ team, and the security server team separately to get an application code release planned and executed. A small support team of multiple products can better provide a single point of contact that helps simplifying the engagement process for technical services, and therefore is a major advantage of this support model.

Last, but not least important, is the transparency of cost. It is easier to define a financial relationship with one infrastructure engineering team. When the cost of infrastructure engineering has to come from many product-based support teams, it is substantially more difficult for the customer to maintain a satisfactory performance management metric that leads to a transparent financial relationship.

Clearly, the infrastructure engineering solution model has the merit of simplicity and the advantages associated with a simple structure. However, with its strengths also lies its major drawbacks. The possible problems of this organization model range from technical training pains to major difficulties in managing large and complex projects.

**Disadvantages of a Support Team of Multiple Products**

Technical training is a big hurdle for a support team working on multiple core technologies and products.

---

6. *WebSphere engineer* is short for WebSphere Application Server system engineer. A WebSphere engineer is an IT system professional who specializes in the design, build, and operation of middleware infrastructures built with the WebSphere Application Server and related WebSphere infrastructure technologies. WebSphere engineers often perform the initial WebSphere system configuration and the ongoing system configuration changes.
Managing the technical training for WebSphere technologies and products alone is a tough job. Managing the technical training well for many core technologies (for example, the WebSphere Application Server, databases, messaging systems, and numerous operating systems) poses a dire challenge for both the WebSphere manager\(^7\) and engineers. For example, the technical manager of the team may not have the kind of knowledge and experience to determine the merits of a large variety of training programs proposed for the group. In addition, there is a limit on how many large sets of complex technologies and products that one system engineer can learn.

Another consideration is training budget. Instructor-led classes are arguably the most effective form of technical training, but this is costly. Such instructor-led training is more cost effective to organize for a large WebSphere organization with teams of WebSphere engineers at strategic locations. Usually, such large classes are charged a fixed fee regardless of the number of students, as long as there are no more than 12 to 15 students. Therefore, they are usually substantially cheaper than sending individual students to different classes. A large WebSphere team can reap the full benefits of the efficiency of large classes and the reduction in travel expenditure. For a smaller team, you may have to send individual engineers to classes at differing locations. You have to pay the full class fee and full travel expense.

As a final point, the size of a large WebSphere organization allows several WebSphere engineers to engage in formal instructor-led technical training together. Such training usually takes a week. For smaller teams, it is much harder to afford the time for technical training. It is not surprising that smaller LOB-based support teams can go for years without any formal technical training.

The technical competence of a smaller engineering organization may become questionable. In addition, this is not only a technical training issue. Focused engineering practice and real-world technical experience are vital in building technical competence. WebSphere technologies, as a subset of JEE specifications, are sizable and complex. Therefore, it is a great challenge from a knowledge acquisition and experience accumulation perspective to learn WebSphere technologies alone, to say nothing of learning many such large and challenging technologies at the same time.

If your engineers have to support WebSphere Application Server, operating systems, databases and messaging systems, the technical competence of your team comes into question because of a lack of focus. Your team may become an organization that can skillfully do entry-level system administration chores, but it is powerless when confronted with difficult system and application problems. Being a junior system administrator in today’s highly competitive IT marketplace may not be an ideal position. Your business partners, your customers, and your senior management are more likely to respect and value a highly technical team that can help them survive a difficult IT job.

\(^7\) A WebSphere manager is a technical manager of a WebSphere Application Server engineering support team.
Finally, recruiting takes more time and is more costly. To hire a good WebSphere engineer is a tremendous challenge anywhere in the world because of the enormous success of WebSphere technologies. The attempt to hire a system engineer who’s good at multiple core technologies, including WebSphere technologies, certainly complicates your hiring strategy, if it’s feasible at all.

**Separate WebSphere Teams**

As previously mentioned, the other choice for a LOB-based support model is to build separate infrastructure engineering teams that are dedicated to WebSphere technologies, but belong to different technical organizations that support one LOB or a large business division (see Figure 1.3.) Separate WebSphere teams belonging to different IT organizations are a step up from the rather undeveloped solution-based support model. It can better deal with technical training, develop in-depth technical expertise, and grow advanced project management capabilities. This section describes some advantages and disadvantages of building separate WebSphere teams.

![Figure 1.3](image_url)  
**Figure 1.3** Separate WebSphere engineering support teams

**Advantages of Separate WebSphere Teams**

A better understanding of the systems and application of the LOB is an advantage that is also true for a solution-based support model. A separate WebSphere team dedicated to one LOB, over time, can develop a better understanding and acquire in-depth knowledge of both the IT infrastructure and the applications. This is because of better focus and less personnel change between the WebSphere teams of a large WebSphere organization.

These organization models make it relatively easier to develop strong work relationships with business partners and the customer. WebSphere engineering is never merely a technical job.
Complex and tough work relationship issues constantly challenge you. As a result, your success is determined not only by how well your team does a set of technical jobs, but also how well you manage your critical work relationships. A support team of multiple products or a separate WebSphere team dedicated to one LOB allows for better opportunities to build good work relationships. The engineers on different teams have an abundance of opportunities to better understand each other working together over a long period of time.

**Disadvantages of Separate WebSphere Teams**

When you decide to build separate WebSphere teams that report to different IT divisions, you have to consider the development stage of your enterprise IT systems. More specifically, you have to ask yourself what is the scale and the speed with which you are interconnecting your mission-critical enterprise IT systems. If your enterprise IT systems are increasingly interconnected, the LOB-based support model may lead to serious issues. This is especially true in the area of practice differences and system inconsistencies. The flaw of this model is the difference in standards and practices. By adopting this model, there may be different WebSphere systems developed and supported by different teams. This is particularly true if you don’t have a centralized WebSphere planning engineering and process engineering function to make the WebSphere Application Server engineering practice consistent. What’s more, it is extremely difficult and costly to correct such engineering differences and the system inconsistencies that have accumulated.

It is costly to support many WebSphere Application Server systems that are fundamentally different. For example, without a centralized WebSphere organization to enforce consistent system standards, different teams design and develop different engineering processes and automation programs for all the different systems. This can lead to enormous redundancy and inefficiency. Differences in practices, processes, and artifacts become established and entrenched with the development of a business division as well as the growth of its IT systems. As a result, it may be costly if your company decides to carry out any process and system consistency effort.

If you decide to correct these differences by introducing system-wide changes (for example, introducing consistent configuration automation), these differences are likely to derail your effort and may result in serious operation errors and production outages.

For example, one WebSphere team may use the same configuration automation program to make WebSphere system configuration changes for all WebSphere environments,\(^8\) including development, testing, and production. However, another WebSphere team may do manual configuration in development and the testing environment while using a configuration automation program to make system configuration changes in the production environment. Thus, using the same WebSphere configuration automation program that your company has decided to deploy to all WebSphere systems, different WebSphere teams may have different experiences and results. The

---

\(^8\) A *WebSphere environment* is a complete set of WebSphere Application Server systems built for a specific purpose. For example, WebSphere production environments are WebSphere Application Server systems built for production operations.
team that uses manual configuration in development and the testing environment may not catch the problems with the configuration automation program. This is because of its established engineering process of not using automation in the testing and development environment. This may likely allow a configuration problem go undetected, slip into the production environment, and cause a serious unscheduled system outage.

These engineering practices differences and system inconsistencies make system integration and system interconnection extremely difficult. System standards, such as server naming conventions, can become deeply rooted second nature for WebSphere engineers after years of usage. Therefore, system inconsistencies, such as different WebSphere Application naming conventions and the different assignment of transport ports, can become serious system stability traps and challenges to integrated or interconnected systems. For instance, in a worst-case scenario, production traffic can be sent to the incorrect destinations because of port assignment standard differences.

WebSphere strategy and process work demand significant resources. A small separate WebSphere team may not be able to afford such resources. The lack of resources to work on critical WebSphere engineering tasks such as standards and processes have a far-reaching impact on the quality of product and service delivery for the WebSphere work in your company. In addition, this lack of enterprise-wide guidance in standards and processes in turn makes the system standard inconsistency and process difference problems become worse and more entrenched, thus enabling a negative cycle of deterioration. It is difficult for a small WebSphere team to support the overhead of planning and process engineering.

Summary of LOB Support Models

For an IT organization of a moderately sized company, both WebSphere organization models may work. However, the significant disadvantages of these models may become overwhelming for large IT organizations.

Typically, large IT organizations are more likely to extensively use WebSphere technologies for critical enterprise IT infrastructure. In addition, these large IT organizations have a stronger need to integrate and interconnect their critical enterprise systems. For this kind of IT organizations, a large and unified WebSphere organization with multiple parallel WebSphere teams dedicated to major LOBs may work better. This is especially true if the overall IT organization is relatively mature with sophisticated project management capabilities, established technology delivery practices, transparent cost model, and experienced technical service delivery management.

WebSphere Organization with Separate Engineering and Service Delivery Functions

For a large WebSphere organization, it is essential to have further specializations. There are natural groupings of talent in the WebSphere organization. Some engineers have the aptitude and gravitate to planning engineering and process engineering work. You can extract planning engineering and process engineering from the actual service delivery. Thus, you have an engineering team and a service delivery team. For these two teams, there are further choices to make.
WebSphere Engineering Team

For a WebSphere engineering team responsible for product strategy and process excellence, you have to decide if this team also works on the engineering work of other middleware products (for example, WebLogic Application Server). There is always pressure to do more with less. Given a limited budget for engineering work, do you want one middleware engineering team that takes on WebLogic, JBOSS, and so on, along with WebSphere technologies? The second question is whether to have a single engineering team that works on both planning engineering and process engineering, or have two separate teams.

Pros and Cons of One Team

The primary benefit of using one team to engineer multiple middleware products clearly is the cost savings. The possible disadvantages of this arrangement can be the lack of focus, competition for resources, and likely delays of key engineering deliverables.

WebSphere product strategy and process engineering are sometimes perceived as “soft” jobs. For example, building product road maps and WebSphere Application Server standards may not be considered as critical or as solid as WebSphere Application Server build or production support. This perception is incorrect. Without top-notch WebSphere talents focusing on an upgrade strategy, WebSphere Application Server standards, and process engineering, your WebSphere organization suffers.

For example, if your WebSphere planning engineer fails to deliver a product road map for your company in a timely fashion, there is no effective guidance in managing a WebSphere Application Server product upgrade. As a result, your product currency program fails. WebSphere Application Server system configuration automation is another example testifying to the importance of WebSphere process engineering. The WebSphere Application Server system configuration is a resource-intensive and error-prone job if done manually because of the existence of many WebSphere configurable items. However, if the process engineers provide the service engineers with a good configuration automation program, the WebSphere configuration work that could have taken days to do can be accomplished in a few hours, if not minutes. For the planners to do the best job possible, a feedback system needs to be built in to the WebSphere organization. Thus, the WebSphere planning engineers aren’t working in a vacuum, but working shoulder to shoulder with service engineers who deliver on the plan.

Virtual Team

The best way to organize a WebSphere engineering team is to have one team dedicated to WebSphere technologies. If this is impossible and your engineering team must support multiple middleware products, encourage your engineering team to build virtual teams with members from the service engineering teams of various middleware products.

This serves two purposes. First, your engineering team has more resources to work on the engineering work for many middleware products. Secondly, it is more likely that the service engineers are more enthusiastic to adopt the standards and processes delivered because they have
been part of the work and consider these the results of their own work. Actually, the virtual team approach may be a good idea even if you have one engineering team dedicated to only WebSphere Application Server technology.

One Engineering Team

Conceptually, WebSphere planning engineering and the WebSphere process engineering need to be considered separately. In engineering practice, it makes sense to have one WebSphere engineering team that do both. This team needs to have no direct production support responsibilities, but it is in charge of WebSphere planning and process engineering deliverables.

WebSphere Service Engineering Team

For the WebSphere service engineering team, you have to deal with whether you want to have separate technical teams to manage system design and system build. At the same time, do you want many parallel service teams responsible for production operations and project work for different LOBs? The project work for these service teams includes planning a build, managing changes, scheduling application code releases, and supporting testing and development environments. Therefore, you can divide WebSphere service engineering function into three teams:

- **Design team.** Responsible for working with the application architect and infrastructure architect to deliver WebSphere topology and configuration design documents.

- **System team.** Responsible for building the WebSphere system and performing system changes and upgrades.

- **Project support team.** Works with the project directly and takes charge of planning, change control, coordination, and WebSphere technical support for production, testing, and development environments.

It is possible to have one WebSphere design team and one WebSphere system team, but multiple parallel WebSphere project support teams for different LOBs.

Should you choose one WebSphere service engineering team that is responsible for the full life cycle of WebSphere product and service delivery? Or will you be better off having separate design, build, project support, and operations functions? When should you have a separate system team and when should you not? The following discussion answers these questions.

Separate WebSphere System Design Team

Having a separate system design team is a great gain for system consistency. It is easier to consistently deliver WebSphere topology and configuration design documents conforming to the WebSphere standards of your company. In addition, a dedicated system design team helps by steadily recommending and influencing best practices in WebSphere systems and JEE application. For example, the system design team may consistently recommend and produce a WebSphere topology that supports redundancy and failover for a critical WebSphere Application Server infrastructure in order to achieve high resiliency.
A possible problem of this team structure is that the system design team does not have to “eat its own dog food.” This team designs a WebSphere system, but it does not have to live with it. In WebSphere engineering, it is mandatory to strive for a balance between conceptual elegance and technical practicality. When the design team does not have to bear the consequences of design flaws, engineering feasibility may become a secondary consideration. Therefore, a separate system design team may deliver something that looks terrific on paper, but is hard to implement in the real world and difficult to support in production operation. A separate system design team may deliver seemingly brilliant WebSphere designs burdened with excessive complexity that is not only difficult to build and support, but is also fragile. In the most devastating cases, the WebSphere system architecture delivered by a dedicated system design team may work so badly in production operation that it has to be replaced by a working WebSphere system via rebuild. This situation becomes worse when the responsibilities of the team are not clear, accountability is not enforced, and feedback between teams is not consistently done.

One of the ways to overcome this disadvantage is to devise the right incentive plan for the system design team to encourage the right behavior. The incentive plan must be linked to the performance of the design. The design team performance could be linked to the following:

- Feedback from the WebSphere system team and the service teams
- Metrics of unscheduled system outages caused by WebSphere system design flaws
- Evaluation of the WebSphere system performance and resilience features against design objectives and requirements

However, having one team provide performance feedback for another team is tricky because it involves the job of managing the sensitive peer work relationship, arguably one of the most challenging relationships in the corporate workplace. In addition, it is not easy to determine that a WebSphere Application Server outage is purely an issue of system design flaws. Often, WebSphere Application Server problems are complex with many contributing factors. Finally, evaluating performance against design goals is easy to say, but it is difficult to do because some of the major WebSphere Application Server design objectives and requirements are not easy to quantify. For example, it is difficult to provide the criteria of WebSphere system stability as a quantitative and measurable design goal. When the designers are not directly involved in fixing the resulting problems, it is hard for them to clearly understand the design issues and the resulting difficulties that the service engineers have to overcome.

**Separate WebSphere System Team for Large Shared WebSphere Environments**

By and large, it is good to have a separate WebSphere system team for large shared WebSphere environments. This team needs to take all of the system work, including system build, configuration change, system documentation, server security, and server recycling. This approach is especially appropriate for mainframe WebSphere system work because high system consistency is important for the stability and availability of these large enterprise servers. Yes, there are sophisticated problem insulation and dynamic system resource management capabilities available on the
mainframe (for example, logical partition [LPAR] and workload manager [WLM]). However, serious WebSphere Application Server problems can still cause widespread stability and availability issues and affect a large number of applications sharing mainframe system resources. Therefore, centralized WebSphere Application Server system work is necessary to enhance accountability, minimize operation errors, and achieve high system consistency for the mainframe.

The process is easy to follow. The WebSphere project team plans system changes and system upgrades and manages the change-control process. However, the project team won’t directly make system changes. Instead, The WebSphere engineers from the project team request system changes through the system team. The WebSphere system team reviews, documents, executes, and certifies the system change. Thus, only one WebSphere team, the WebSphere system team, makes changes to the large shared WebSphere environments and is held responsible for the changes made. This is infinitely safer from a system stability perspective than having multiple WebSphere technical teams work on the same large shared WebSphere environments.

For large shared environments, it is critical that, at any given time, one WebSphere team has a comprehensive view of the entire WebSphere systems and is accountable for system consistency. Without such a centralized authority in WebSphere system work for large shared environments, it is difficult to maintain system consistency. As a result, your system is not stable. This is especially true for large WebSphere systems hosted on enterprise servers, such as mainframe servers.

Mainframe technical training and technical skill considerations support the arrangement of a separate WebSphere system team. You could have a small group of mainframe system experts to form a WebSphere system team. Meanwhile, many WebSphere engineers with JEE exposure and project management capabilities could comprise WebSphere project teams. This organization model makes your teams more effective and technical training easier.

Sure enough, there will be communication, cooperation, and collaboration issues between the WebSphere system team and the WebSphere project teams. From time to time, there will be a need for all the WebSphere teams to focus on improving teamwork and learn how to better work together on these large shared WebSphere environments. However, this is a necessary price to pay. Considering the system consistency and the stability advantage of this organization model, this is a balanced approach for these important large shared environments.

However, for large WebSphere environments built on a distributed platform that is dedicated to a single large and complex IT project, it is altogether a different issue.

**Separate WebSphere System Team for Dedicated Environment on Distributed Platforms**

A separate system team can be a good organization choice for a large shared environment, such as mainframe WebSphere Application Server systems. For WebSphere systems built on distributed platforms for large and complex IT projects belonging to one LOB, a separate WebSphere system team may not be necessary. For example, for the mainframe, the WebSphere Application
Server build requires specific system knowledge of the mainframe that takes years to learn. Therefore, a dedicated team that has the right staff can do a better job at a system build. For distributed platforms, a system build is a straightforward job that an average WebSphere engineer can perform.

This separation of system work may add an extra layer of service delivery dependency for the WebSphere service team. Your project team now has to take on the extra coordination with the system team to deliver any system work without extra benefits:

- The WebSphere project team is completely able to build servers.
- This is not a large shared environment where tight control of the system is necessary to provide system stability for many applications, but a dedicated environment.
- There is only one WebSphere team working in this environment.

Assigning a separate WebSphere system team to do system work on the distributed platform, you may have to deal with some issues.

The lack of clear division of labor may become the first troublesome area. Division of labor can easily become an issue between your WebSphere system team and your WebSphere project teams. They all belong to your WebSphere organization with a shared budget, workload, and human resource distribution across teams, but which team should be responsible for what tasks? Should the system team deliver a completely functional WebSphere Application Server to the service team? If there are any system issues during a major upgrade, should the system team be held accountable to fix the problems? It is critical that the division of labor between teams be one of the first decisions made so that everyone is on the same page.

Delays in system work are not uncommon. This is because there is one WebSphere system team with many WebSphere project teams competing for its resources and priority, as well as the resulting scheduling and coordinating challenges. Chapter 10, “System Upgrade and Product Maintenance Management,” provides detailed discussion on the difficulties in competing for resources and scheduling for system work.

For distributed platforms, this model of separating system work from the project teams may cause problems between the WebSphere teams involved. This type of relationship issue within the same WebSphere organization is typically tough to deal with, even for experienced WebSphere managers and senior consultants.

Is this a core organization issue or a challenge to the management of technical teams? Will the benefit of centralized system control, consistent system build, and a high level of system consistency that typically comes from a separate system team justify the extra overhead, coordination, and organizational issues? Can a separate system achieve a higher level of automation? Different WebSphere professionals with different experiences and backgrounds may have different opinions. This book does not advocate one organization model against another. Rather, it is an experienced observation coming from long years of hands-on WebSphere engineering practice. Even for experienced practitioners, it is important to keep an open mind, be ready to adapt to the
changes of business drivers, embrace positive changes, and design and implement technical organi-
ization according to the business actuality of your company.

Of course, nowadays, any discussion of building a high quality WebSphere organization is not complete without figuring out integration strategies and the support models of global deployment.

Building a Global WebSphere Workforce

An open mind and a positive attitude are needed to address the substantial disparity between financial resources available and the quantity and quality of work required for large IT infrastruc-
ture engineering organizations worldwide. Determined efficiency and productivity drive help, but they do not alter the resource-intensive nature of IT. IT is still in its infancy. It will take time before IT matures to an adequate level of industrial strength standardization and engineering process automation. Only then can the IT industry significantly reduce its dependency on high human resource consumption. Developing a global technical workforce has been one of the responses to financial and quality challenges. Globalization is a fundamental sourcing strategy, as well as an attempt at product quality and service delivery improvement.

Integrated Team Model

The integrated team model, or mirror team model, implements the concept of global teams incor-
porating onshore and offshore technical and management resources as each team sees fit in terms of the nature of the work, as well as time and location.

The blanket offshoring approach that hollows out the technical work of a large IT division seems more suitable for application development. Instead, the integrated team model for building a global technical team seems more appropriate for an organization responsible for enterprise infrastructure technology, such as WebSphere Application Server engineering.

The management and technical talents of a large enterprise infrastructure engineering organi-
zation belong to the core assets of a large company that has sizable and complex IT infrastructure essential to the continued success and prosperity of the company. The stability of the infrastructure engineering teams, such as support for critical WebSphere Application Server technology, has a direct relationship to the stability of the company’s key IT infrastructure.

The integrated team model approach presents the least disruption to the WebSphere organi-
zation and the critical WebSphere Application Server infrastructure. It is important to have mini-
imum organizational and functional disruption while building a global technical force. IT infrastructure engineering and system operations are different from application development. The infrastructure problems tend to be more direct, immediate, and pervasive. As far as application development is concerned, there are usually rigorous testing activities between development and production. Testing, especially stress testing, uncovers and eliminates application code and system configuration defects. Normally, no application code directly migrates to production envi-
rionment without testing. However, there is nothing standing between your most critical produc-
tion WebSphere Application Server systems and system operations that your WebSphere
engineers perform. For example, the accidental launch of a WebSphere configuration program can incorrectly reconfigure one of your major WebSphere production systems, render it completely unusable, and cause a tremendous enterprise-wide outage. For enterprise infrastructure, especially for large and key technical environments, any disruption is instant, widespread, and serious. The integrated team model helps minimize such problems.

For the offshore WebSphere engineers and managers to be successful, it is critical to help them do a good job at technical training, learn the environments, and know the business partners. One of the best ways to achieve these objectives is to assign them to different WebSphere teams and work with the team for an extended period of time—the minimum needs to be one year. When they know the WebSphere systems, applications, engineering processes, and various technical teams and business partners, it is the right time to reconsider their assignment by fine-tuning the global organization. The integrated team model allows the best opportunity to train offshore WebSphere teammates and give them the best chance to be successful.

This integrated team model gives a WebSphere organization better control and flexibility over sourcing options, as stated here:

- An organization has total control over the scope and degree of this practice and can grow and downsize either onshore or offshore components of the WebSphere organization promptly according to the change in business drivers.
- In addition, this approach gives a better opportunity to directly manage the quality of WebSphere engineers both onshore and offshore via established hiring practices and professional and technical training mechanisms and channels.
- This model allows maximum control over the assignment of the roles and responsibilities of a global team with its onshore and offshore team members.

### Long-Term Sourcing Strategy

Building a global workforce must not be taken only as a convenient means of labor arbitrage. Offshore resources will play an increasingly indispensable role in balancing labor costs, accessing rare skills, improving service quality, and reducing service latency.

For example, it is unlikely that all global locations will have exactly the same economic cycles. During a boom time in the U.S., less expensive and relatively more readily available WebSphere resources at offshore locations may be employed at greater numbers to reduce the overall labor costs for your WebSphere organization. In difficult moments of grave budgetary constraints, substantially more WebSphere engineering services can be performed at low cost by hiring more workers with good WebSphere skills at offshore locations. The offshore operations must play a balancing function in the overall WebSphere infrastructure business.

---

9. WebSphere infrastructure refers to WebSphere technologies centered on IT infrastructure. These WebSphere technologies primarily include WebSphere Application Server, WebSphere Enterprise Service Broker or WebSphere Data Power, and others. WebSphere infrastructure specifically refers to the physical servers, network connectivity, load-balancing devices, security mechanisms, operating systems, and WebSphere system software installed, among other infrastructure components.
As a long-term strategy, the cost savings of offshore labor should be secondary. The focus of building a global WebSphere Application Server engineering organization should have other more sustainable objectives. The gaps in salary for onshore and offshore staff and will rapidly disappear. With the extra cost of managing a sizable offshore staff, there could be more costs rather than savings.

As a long-term benefit of globalization, a 24/7 support model is certainly worth exploring. A global WebSphere team and its members at different geographical locations are complementary to each other in engineering support and system operations. For example, China and India have different time zones from those of the U.S. with an 8- to 13-hour difference, depending on the time zones chosen for comparison. This time difference enables a true 24/7 nonstop support system operation, thus reducing human errors frequently associated with late-hour technical support. This helps cut down unscheduled system outage and improves the overall quality of service. The time difference can also reduce overtime payment that a company must make to be compliant with various state and federal labor laws and regulations at applicable locations and further reduce the cost of operations.

Building a local IT organization at strategic high growth locations, such as India and China, can be advantageous, too. This lays the necessary technical organization foundation to support the global growth visions of the company.

In the long run, managing project work and accommodating the real-time nature of infrastructure engineering support is key to the success of a WebSphere team of global workforce.

The WebSphere Application Server engineer has a project management perspective. A majority of change-management meetings, project meetings, and team huddles often occur during normal business hours. The primary WebSphere engineer must attend these meetings. It remains to be seen how this type of work can be effectively managed by global resources.

What’s more, to be an effective primary WebSphere engineer, being able to perform real-time engineering or operations support is critical. The majority of such operations are during normal business hours, rather than at night.

Organizing global resources to support the above mentioned project work and real-time engineering activities and operations eventually determines the size of the global WebSphere team and the amount of assignments for the global team. One obvious solution is to get two or three shifts for the offshore teams to rotate into project work, real-time engineering support, and system operations during local night time. However, hiring and retaining WebSphere engineers anywhere in the world is becoming a great challenge. For some strategic global sites, such as India and China, the competition for WebSphere engineers is white hot. Beijing recruiters have even reached out to the U.S. market for senior WebSphere talents. Any WebSphere manager of global teams knows too well how difficult it is for hiring and retaining WebSphere engineers anywhere in the world. In a tough marketplace for hiring and retaining WebSphere engineers, requesting team members to take night shifts may not be the best way to build and keep a global WebSphere team.
From the perspective of building and managing global teams, there are many interesting and significant organization questions: how far an IT infrastructure engineering team must go to expand a global workforce, what the right balance of onshore, nearshore, and offshore teams should be, and how to distribute the project, engineering, and system support work across the globe.

Although there are no cookie-cutter approaches and there is no one-size-fits-all solution, it is critical that a global WebSphere team be able to engage in end-to-end WebSphere Application Server engineering support, from engagement and design to production operation and decommissioning.

One approach to overcome the difficulty of globalizing a WebSphere organization is to abolish the model in which the WebSphere team assigns a primary WebSphere engineer and secondary engineer to a large project during the full engineering life cycles of the project, and assign a service request system that reaches out to a pool of WebSphere engineers for support. Modern IT projects involving WebSphere Application Server are typically large and complex. Engineering reality is the best test bed for any organizational approach. It remains to be seen if such a pooled WebSphere engineer structure will meet the needs for the timely delivery of quality WebSphere Application Server products and services.

**Engineering Support with a Three-Level Approach**

A technical support model with a three-level structure has been popular with application development and support organizations. This technical support organization model has also been adopted by some technical teams supporting IT infrastructure. For example, some operating system support organizations may adopt a three-level resource pool support structure. However, what makes sense for some IT infrastructure engineering groups may not make sense for the WebSphere organization. This is because WebSphere technology is a part of JEE specifications. This relationship ties the WebSphere Application Server system tightly with the JEE applications that execute within WebSphere JEE containers. This close interdependency between the WebSphere Application Server system and WebSphere applications makes it necessary for the WebSphere team to carefully consider the impact of its support structure and organization choice over the application teams, especially the application support teams.

The shared responsibility of the WebSphere team and the application support team for supporting the WebSphere application and the initial uncertainty of the team responsible for the root cause of a production problem requires that the two teams work well together.

10. A WebSphere application is a JEE application designed and developed to execute within a WebSphere Application Server.
11. The application support team is also called the production support team. This technical support team ensures that the application is in stable production operation. This team has expert knowledge of the application operations. It works closely with all the technical teams, such as the WebSphere team, application development team, and testing team. The application support team is the first line of defense against possible stability problems. During production problem resolution, the application support team usually plays a leadership role.
Level 1 Technical Support

The application support team plays a pivotal role in both production application support and system stability. The application support team is the first line of defense against system stability issues. It has in-depth knowledge of the complex mission-critical application for which the WebSphere team provides a production execution environment. It also has limited WebSphere Application Server system privileges to expedite application production support chores.

For example, the application support team may have operator system privileges for the WebSphere Application Server. As a practical consideration, the WebSphere team most likely would want to grant operator privileges to the application support team because this reduces the WebSphere team participation in production issues that the production support team12 is fully capable of managing. (It is not unusual to see the WebSphere team provide or share WebSphere technical training opportunities with the application support teams. Chapter 11, “Critical Work Relationships,” discusses this topic.)

In this situation, the application support team calls in the WebSphere team when it has a difficult problem that demands expert-level WebSphere Application Server knowledge and skills, and the WebSphere team plays the role of Subject Matter Expert (SME), consultant, and senior-level technical support.

If the application support team is already playing the limited function of WebSphere technical support, installing a WebSphere Level 1 technical support role most likely creates a parallel or overlapping function with the application production team.

First, the WebSphere Level 1 support team may not have the necessary application knowledge to tell whether there is an application issue or a WebSphere system problem.

Second, when there is a difficult WebSphere system problem for which the production support team needs to call in WebSphere technical support, the WebSphere Level 1 team support may not be able to provide the senior-level WebSphere technical expertise needed. The WebSphere Level 1 support team may have to pass it on to the next level without adding value.

A third concern is the speed with which the right kind of WebSphere technical assistance is reached. When the company suffers a production issue, the application production support team does not have the patience to call in WebSphere Level 1 technical support when the only assistance provided may be sending out a page to WebSphere Level 2 support, who are capable of providing the kind of WebSphere technical assistance needed. When the WebSphere team is called upon to help, it is usually a serious and difficult problem that needs immediate attention. During a production emergency, it may be irritating to all the technical teams involved if there are any delays engaging the right WebSphere expert.

12. Also called the application support team or production support team, the application production support team provides advanced technical support for applications. This team has in-depth understanding of applications. It is the first line of defense for the system stability and smooth operation of the application. This team coordinates triage and problem resolutions when a production problem occurs, and it works with application development teams and other technical teams to provide a permanent fix for production problems. This team may have limited WebSphere administration privileges, which usually include recycling JVM instances and browsing WebSphere log files.
A large company may have thousands of large and complex WebSphere Application Server systems from different business divisions. It takes time to learn only a subset of these systems and become effective in providing support. A large IT organization is seldom a perfect world of ideal process optimization and flawless system consistency. Instead, large and complex WebSphere Application Server systems have a long history of evolution through mergers, acquisitions, business changes, and organization dynamics, and as a result, system differences and process inconsistencies exist. The WebSphere team must work in such reality. Therefore, the fourth area of concern is the difficulty in learning a large number of complex systems. In such a challenging environment, it is difficult for each engineer to provide WebSphere technical support for all the WebSphere systems and applications of a large company. This approach has pronounced technical training difficulties. It may not be possible for one WebSphere engineer to learn a large number of complex WebSphere Application Server systems.

**Level 2 Technical Support**

WebSphere Level 2 support has a solid set of WebSphere Application Server engineering tasks to perform, such as, deep dive performance tuning, troubleshooting, and high-impact and high-visibility production problem resolution. This job can cover the complete category of WebSphere Application Server engineering and it is a stable, visible, and desirable position.

If you have a WebSphere Level 2 technical support role, the best practice needs to be building a technical Level 2 support team with team members at each of the major geographical locations. This allows you to build a truly global WebSphere team that can provide the best WebSphere support around the clock. For example, you have the best engineering support around the clock if you have WebSphere Level 2 support around the world rather than concentrating on one geographic location. In addition, even global talent distribution makes it easier to build a local career path and mentoring opportunities for the WebSphere organization.

**Level 3 Technical Support**

The WebSphere Level 3 technical support role may have an overlapping role in terms of its function, especially for difficult production problems with several teams, and poses additional management issues. For example, if the WebSphere Level 3 support team works on application code to isolate defects, is that a service defined in a service level agreement (SLA) and, if not, what is the financial arrangement between organizations? If the WebSphere Level 3 support team provides application code improvements to fix a problem, who should be responsible for production instability introduced by the defects in the code improvements provided?

Another case is how to escalate and engage the IBM WebSphere Technical Support Organization when you have production problems and WebSphere Level 3 technical support. If a WebSphere Level 3 technical support role is set up, it is necessary to define the work relationship of
the Level 3 support with IBM technical support, and how to engage IBM for technical support. The following exercise may help determine whether a WebSphere Level 3 technical support role is required.

As an established practice, IBM WebSphere Support and Java Support Organizations usually play the role of advanced WebSphere technical support. As with obtaining any technical support, there is an overhead cost in engaging IBM WebSphere and Java support. An IBM Problem Management Request (PMR) has to be opened with a detailed description of the technical problem that is occurring. Then, depending on the severity and difficulty of the problem, many technical discussions with IBM Level 1 or Level 2 WebSphere support or IBM Java support (or both) will likely take place.

If you have three levels of WebSphere technical support, it is an interesting question as to when and who needs to open an IBM PMR to secure IBM technical support. Let’s assume that the Level 1 WebSphere support won’t bypass Level 2 and Level 3 WebSphere support and therefore the Level 1 WebSphere support does not open an IBM PMR to seek IBM technical support. Let’s say that the Level 2 WebSphere support engineer looks into the problem, but determines that more advanced WebSphere technical support is needed. Then, what does the Level 2 WebSphere support engineer do for an urgent production problem? Escalate to the WebSphere Level 3 support or open a PMR and seek technical support from IBM? Say that the technical support process stipulates that your Level 2 WebSphere support must escalate to the WebSphere Level 3 support, and the Level 2 support follows the process and escalates the problem to Level 3. In that case, the urgent production issue has already traversed, at this point, through three layers of technical support within a WebSphere organization. With each layer of technical support hand-off, there is a time-consuming burden of communicating a complex technical problem. Note that this is at a moment when the WebSphere engineers need to focus on solving a tough technical problem under the pressure to stabilize the WebSphere Application Server system and restore production operation. If the Level 3 WebSphere support decides to seek IBM WebSphere support, the WebSphere Level 3 support has to spend time both opening a PMR and discussing the problem with IBM Level 1 or Level 2 technical support. Figure 1.4 describes this support model.

Of course, to avoid a WebSphere technical support request traveling through five layers of engagement and communication process in order to loop in all WebSphere experts, and more importantly, to solve the production problem as soon as possible, the WebSphere Level 1 support and the application production support can immediately engage both the WebSphere Level 2 and IBM WebSphere support, as well as the WebSphere Level 3 support. Then, should this structure of three levels of WebSphere technical support be flattened? Does the WebSphere Level 3 support function overlap with IBM WebSphere support, the same as the WebSphere Level 1 support function overlaps with that of the application production support?

In the tough world of WebSphere engineering, where production problems urgently demand many technical teams promptly working together for a timely solution, an organization structure with many layers of laborious support escalation process is cumbersome and impractical.
Figure 1.4  Three-level structure of WebSphere technical support

JEE experts in the application development organization need to also be considered. Because WebSphere technology is an implementation of a subset of JEE specifications, it is helpful to have assistance from senior JEE experts. This is especially true when there are production emergencies and there is a need to better understand the JEE application. WebSphere Level 3 support must have solid JEE experience. However, senior JEE developers and application architects in the application development organization can and should play the role of JEE application consulting during a production emergency.

The JEE experts on the WebSphere team are focused on the infrastructure side of the middleware work while the JEE developers and architects spend more time on the JEE applications. The developers and architects of the JEE application certainly know more of their application design philosophy and the technical details of their application code.

In addition, in terms of the overall IT organization, it is more efficient and cheaper to build a temporary task force to solve a serious technical problem, rather than training, retaining, and paying for WebSphere Level 3 support engineers for their advanced JEE application development expertise, which is not heavily used.

WebSphere plan engineers and process engineers are usually senior WebSphere professionals. They can be mobilized to participate in difficult technical problem resolution as part of a task force during a high-severity production incident. This is a more resourceful organization approach than keeping a pool of senior WebSphere engineers as Level 3 WebSphere support.
WebSphere technical support with three dedicated levels of expertise may look elegant and sound logical. It may look like a good way to organize because it has clear-cut divisions according to the skill levels. It could appear to be a logical way to separate WebSphere production support work based on its perceived efficiency of using the right talent for the right job. However, this three-level resource pool support model will most likely be unable to endure the test of time and engineering practice to provide unified support for a large company with a sizable WebSphere Application Server installation base that holds numerous complex WebSphere applications:

- There are technical training difficulties with learning a large number of complex WebSphere systems, practically all the WebSphere applications and infrastructure that a company has. This is especially true for a large company with a large number of WebSphere Application Server systems.
- Possible overlapping responsibilities and redundant roles with production support and IBM technical organization when the roles and responsibility, as well as engagement model, are not well defined.
- Reduced speed in production service restoration and production problem resolution, especially when the engagement and escalation process is not optimized.

It helps to have a flattened team structure with normal on-call rotation for every WebSphere engineer. This allows all the technical talents, including IBM, the WebSphere engineers, and the application development team, work shoulder to shoulder as an effective technical force to resolve tough problems together.

Nevertheless, a carefully defined process and a clear division of roles and responsibilities may help mitigate the risk of a layered support model. For example, problems can be classified according to difficulty and severity. For low-severity and relatively less difficult problems, Level 1 support can work independently. For high-severity problems, Level 2 and Level 3 need to be informed of the problem and provide leadership and guidance to Level 1, or participate directly in the problem resolution when appropriate. Therefore, the Level 1 WebSphere support performs the usual support chores, such as recycling the server, collecting and uploading data for IBM, opening a PMR, documenting the problem, and providing a report to senior management. The Level 2 and Level 3 support can play the leadership role in problem resolution and work as consultants and mentors to Level 1 WebSphere support. A flexible tiered structure of a virtual team and rotation may also work. For example, senior WebSphere planning engineers and process engineers can serve as Level 2 and Level 3 support when needed. For example, when the WebSphere team fights through a major production emergency, senior WebSphere engineers can be pulled in to help Level 1 WebSphere support engineers and work together as a virtual team.

A leveled resource pool model may be suitable for supporting a moderate number of WebSphere Application Server systems because it is possible for the WebSphere team members to learn a reasonable number of large systems, such as, supporting the WebSphere Application Server systems of one business division.
This model works well if the WebSphere Application Systems are standardized and the WebSphere applications are consistent in terms of architecture and operations. For example, a shared environment that holds many WebSphere applications belongs to one business division.

**Technical Support for Very Large IT Projects with Multigenerational Plans**

WebSphere Application Server technology is typically used to host powerful but complex JEE applications. Some of these large IT projects have multigenerational plans. There will most likely be active new initiatives for years. To support such large IT projects, a level of support personnel stability is necessary. This helps your team deal with complex technical details, maintain project management continuity, and build work relationships with peer technical teams and business partners.

Let’s look at what you can do to best assist the previously mentioned large IT projects.

**Building a Small Subteam**

These large IT projects are the manifestation of senior management’s determination and the financial commitment of an IT organization for top priority as well as strategic business initiatives. Besides, these are difficult projects because of size, complexity, and visibility. You want to carefully choose those who are most suitable to be the members of a small, but elite, WebSphere subteam.

Of course, you want to use your best and most experienced WebSphere engineers for the most important WebSphere projects. However, they are scarce resources. You may also want to pair off senior WebSphere engineers or consultants with junior team members. This allows your senior WebSphere engineer or top consultant to delegate entry-level technical or project management tasks to junior team members. This allows your senior team members to focus on complex and difficult work and add more value. Your junior engineers have an opportunity to learn to do a job, build confidence, and gradually take on more challenges under the guidance of senior team members. This arrangement also helps you to better deal with possible staff turnover and provide continued quality support for your strategic projects.

**Separate WebSphere Support**

For the WebSphere engineering support of a very large IT project, you have the following two distinct areas of focus:

- Production environments
- Testing and development environments

It is important to understand the specific requirements for each area and assign the right kind of WebSphere engineers and consultants.
For production environment support, the WebSphere engineer or consultant selected must have good technical skills. However, equally important, if not more critical, are good communication skills and project management capabilities.

For the selected WebSphere engineer to lead the WebSphere work in a large production environment, she must be able to communicate clearly and powerfully. A factual, considerate, and assertive manner is highly desired. Key communication capability is particularly important when working with the production support team on resolving differences during production emergencies.

The second attribute of the role is the ability to have both a great sense of urgency to get the job done while keeping a calm and professional demeanor. For example, the team members must focus on resolving a nagging production problem and restoring the service of an important production system while confidently and professionally interacting with the team and management.

Last, but not least, is project management capabilities. The candidate for this senior WebSphere engineer position must understand the value of carefully managing a large WebSphere project. She needs to be familiar with the practices and mechanisms of project planning, change management, job scheduling, quality assurance, and progress management. For example, the candidate must be able to lead the subteam to deliver many servers within deadline. The candidate must have an appreciation of quality control (for example, diligently following established server build and validation processes). The candidate must have the enthusiasm to perform a large number of difficult project management jobs. For example, she must have the patience to participate in lengthy change-control meetings, and must fight through planning sessions for the right window to perform a critical change. Without technical project management skills and enthusiasm, success is elusive for the candidate.

The next area of focus needs to be the testing and development environment support, including the critical stress-testing environment, which requires a different type of WebSphere engineer. Some of the attributes for this role include JEE development background, enthusiasm for deep diving into technical niceties of tough problems, and the patience of combing through large amounts of testing data to reach recommendations for solving tough system problems or complex performance issues.

JEE development experience comes in handy for collaborating and conversing with testers, developers, and architects to resolve difficult technical problems uncovered during testing—especially stress testing.

Curiosity and interest in understanding and tackling tough technical problems are essential for this role. The WebSphere engineer assigned to this job must not be someone who only has the technical knowledge and skills needed to do the job. A suitable candidate must truly enjoy solving tough technical problems and be proud and excited to provide excellent solutions.

Patience for combing through a large amount of testing data is important. The WebSphere engineer must have the patience to work with testers to analyze and research testing results. The
WebSphere engineer must have the perseverance to participate in repeated tests and analysis to seek optimal system configurations.

The candidate must have exposure to a broad set of technologies. This helps her develop an end-to-end view of a large WebSphere Application Server-centered IT system with many integrated components and interconnected systems. In particular, this is important for optimizing WebSphere Application Server systems in problem avoidance and performance enhancement. For example, from a traffic and load perspective, an end-to-end view from the customer browser, geographical load balancer, Web server, security server, and application server, all the way to the backend enterprise data store helps in constructing a fine-tuned traffic pipeline that minimizes system problems and maximizes system performance.

Ensure Standards and Practice Consistency

Over time, a large IT project can evolve into different system standards and engineering practice. In the long run, such differences are costly to correct, if possible at all. Therefore, you need special countermeasures against possible consistency problems for large projects. Both organization adjustments and engineering processes can help reduce the inconsistencies.

System Consistency Challenge

Having a small team engaged on a large WebSphere project for an extended period of time may present unique challenges. One of the challenges is deviations from WebSphere Application Server standards, which would then cause inconsistencies in engineering practices. For example, to expedite a WebSphere Application Server build to suit an application, the WebSphere engineer engaged may define the resources, such as Java Database Connectivity (JDBC), at server level rather than at node level, as your enterprise WebSphere Application Server build standard recommends. This may lead to a problem when a different WebSphere engineer updates the resource definition. He may be unaware that a nonstandard resource definition was used. As a result, the change in resource definition may be made at the node level rather than server level. This causes the application to fail when it tries to find the resource definition, and an unscheduled production outage occurs. It also causes automation problems that are to be used to audit enterprise systems.

Team WebSphere System Architect

To prevent such inconsistency problems, you can assign a senior WebSphere engineer with both WebSphere Application Server system experience and JEE expertise to work with all the projects as the team system architect. The primary role of the team system architect is not to design every possible WebSphere Application Server topology and configuration document for all the projects for which your team is responsible. (Of course, the team system architect can always provide input for WebSphere Application Server architecture issues.) However, his primary function is to review every WebSphere Application Server topology and configuration document that your WebSphere team delivers to ensure that the documents conform to the enterprise WebSphere standards and generally accepted WebSphere engineering practice.
System Audit
Another effective means to ensure WebSphere Application Server system consistency is to periodically conduct a system audit. This audit can be done using an automated program and, when necessary, a manual process. For example, a quarterly WebSphere system audit can be done. You need to also perform a spot check of the system as an ad-hoc quality assurance. Any problems uncovered during the system audit need to be corrected. WebSphere system audits without rigorous follow-up are useless.

Rotation and System Stability Considerations
A reasonable partial rotation of WebSphere engineers for large and important IT projects can help achieve high WebSphere Application Server system consistency and standardized engineering practices. However, this rotation may also destabilize your critical WebSphere Application Server systems and disrupt work relationships. These large IT projects need organization stability to be successful. Therefore, you have to carefully weigh the pros and cons and reach a balanced solution to any intended rotation as a correction to possible system inconsistencies. Of course, there are always many organizational considerations, business objectives, and project imperatives that you need to consider with personnel changes.

WebSphere Center of Excellence
The WebSphere Center of Excellence (WCoE) can be a virtual organization that is used as a forum for WebSphere technology discussions to help build consensus in your IT organization. WCoE can also be a full-scale technical organization with a clear emphasis on consulting rather than direct product or service delivery to the customers. Either way, a good WCoE is useful and valuable.

Virtual Organization
When a WCoE functions as a virtual organization of WebSphere Application Server technology, its function is similar to that of WebSphere planning engineering. However, it has a different emphasis on providing a forum for technology discussion in order to build consensus. In addition, the members can work together to review and approve product strategies, plans, and roadmaps. For appropriate cases, WCoE can make infrastructure technology recommendations, such as the following, to the senior management for approval:

- Evaluate and approve target legacy systems and form conversion strategies.
- Evaluate and approve migration strategies, plans, and roadmaps for WebSphere technologies.
- Evaluate emerging technologies and make recommendations on introduction strategies and plans.
- Evaluate JEE standards development and approve migration strategies and plans.
As far as existing products (for example, WebSphere Application Server), WCoE needs to be able to discuss and approve product strategies, plans, and roadmaps. For new products, such as WebSphere Process Server, WCoE can facilitate discussions and make recommendations to the enterprise infrastructure architectural authority for approval or disapproval about the use of the technology in a company. In other words, the WCoE needs to be an ongoing meeting that is a forum for stakeholders to review new technology and best practices.

This kind of WCoE is a critical organization where you can exercise influence over the technological future of the WebSphere technologies in your company. WCoE’s work can shape the future for your WebSphere teams. You want to lead and participate in making key decisions working with the other WCoE members.

This is a virtual team that you want to be part of. This is a forum that you do not want to miss.

**Full-Scale WebSphere Engineering Organization**

WCoE can be a full-sized WebSphere Application Server engineering organization with an emphasis on WebSphere consulting rather than direct product and service delivery. As seen in Figure 1.5, WCoE has a unique organization structure:

- JEE standards and framework
- WebSphere performance testing and monitoring
- WebSphere operations excellence

WCoE needs to be staffed with experienced WebSphere engineers whose job is to help other WebSphere engineers be more successful. These senior WebSphere engineers do not directly provide WebSphere Application Server products and service to the customers. Instead, they provide engineering artifacts, such as monitoring strategy for performance testing and automation scripts for system audit. They are also pulled into production emergencies to help solve tough technical problems, or are otherwise engaged to assist in resolving for difficult technical problems. However, WCoE does not have day-to-day production on-call responsibilities.

**JEE Standards and Framework**

Senior JEE developers and architects will be part of WCoE. They will work with the application development organization to deliver JEE application standards and JEE application best practices. They will be responsible for the design and implementation of JEE application framework. All JEE applications with WebSphere as the target deployment environment must use the framework in application development.

These JEE experts provide consulting services for JEE standards, application framework, and WebSphere application best practices. This consulting service is not only for other WebSphere engineers, but also for enterprise architects, application developers, and software vendors. If this WCoE team does a good job upfront, many potential application and system problems are eliminated in preemption (for example, before the software defects have a chance to be programmed into the applications).
WebSphere Performance Testing and Monitoring

WebSphere performance testing and monitoring strategies are critical to the quality and stability of the systems delivered and managed. Consultants who focus on performance and monitoring work are both WebSphere infrastructure and IT infrastructure experts. These WebSphere performance and monitoring consultants have extensive knowledge of the IT infrastructure of a company. They know how many enterprise applications are interconnected or integrated through the system capabilities provided by the infrastructure.

Thanks to their extensive infrastructure and application knowledge, WebSphere performance and monitoring consultants know where, how, and what to monitor in production and performance testing environments. They know how to retrieve detailed performance data in target areas. They use these precious performance data to help the testing organization form highly effective testing strategies and plans. In other words, they are the experts of the powerful methodology of selectively monitoring production systems. They retrieve relevant performance data to guide corresponding performance testing. They know how to use the performance data harvested in testing systems to further hone the production environment. In addition, they have the low-level technical skills to help them perform their job and achieve their technical vision. For example, they know how to create or use system tools to monitor various infrastructure components, such as networks, OSs, JVM, and Java applications.
This team is the high-strength glue that powerfully connects production and testing. These WebSphere performance and monitoring experts are fearsome warriors who neutralize system and application defects with deadly accurate “smart bombs.” They are mighty defenders of your WebSphere system stability and availability, working with your testing organization and application support team.

**WebSphere Operations Excellence**

WebSphere operation excellence is all about quality automation that is tested and well controlled. This is presided over by WebSphere engineering process and automation experts. They are operations specialists whose job is to help the service engineers reach a high level of productivity and delivery quality through process automation. Their charge is to design process automation programs and apply script programming skills to develop the automation programs.

These WebSphere engineers also provide process automation consulting and help the service engineers customize the automation programs. The team helps to substantially improve the productivity of a team while reducing many human errors in system operations.

Production support refers to participating in production-related engineering and operations support, such as providing production with on-call support. WCoE engineers must have no direct production on-call support responsibilities, but can be engaged for difficult and critical problems as SMEs.

WCoE, as a full-blown WebSphere engineering organization, can belong both to a product-based support model or solution-oriented support model. WCoE can be the engineering division of a large WebSphere organization, or provide support to solutions-based teams.

WebSphere service engineering teams assigned to different LOBs can collectively function as a peer organization to WCoE. WCoE and WebSphere service engineering teams work together on WebSphere planning engineering through a virtual team led by WCoE.

**Summary**

Building a WebSphere organization for a large company is an exceedingly complex job, especially in today’s challenging business environment of globalization amid fast-changing technologies and constant business dynamics. There always seems to be a large number of seemingly contradictory and competing factors and objectives.

For example, for a technical organization to be highly effective and competitive in WebSphere technologies, it needs to build a product-based WebSphere organization for technology focus and technical specialization. However, product-based support models present engagement and coordination challenges, such as the lack of a single point of contact and the need for a specialized team to help with system maintenance and change coordination among technical teams.

Another example is the separation of WebSphere planning engineering and process engineering from WebSphere service engineering. If an organization has this separation of WebSphere engineering functions, the WebSphere planning engineers and WebSphere process
engineers are not distracted by daily production concerns; therefore, they can focus on delivering important WebSphere engineering artifacts. However, this separation, if not managed appropriately, may lead to the creation of WebSphere standards and processes not usable in real-world WebSphere engineering practice.

Most importantly, you must consider your overall IT organization and the engineering and process reality of a company. No WebSphere organization model can solve the systemic issues of an IT organization. You have to understand how far you can go and what you can do within your company’s specific situation.

However you choose to organize your WebSphere organization, the bottom line is clear: It must help build a highly technical engineering team, enhancing accountability, and improving the quality of WebSphere products and service delivery.

Sun Tzu said, “Anciently the skillful warriors first made themselves invincible and waited for the enemy’s moment of vulnerability.” This quote reveals the intent of this chapter: optimally organize a WebSphere organization, preferably a large infrastructure engineering support organization dedicated to WebSphere technologies. By doing so, an organization positions its WebSphere teams in a winning alignment.

Now you have an organization framework with which you can start building a world-class WebSphere technical force of unsurpassed performance, a topic that is discussed in Chapter 2, “Building a World-Class WebSphere Team Through Hiring and Training.”
Index

A
activities, 82
activity diagrams, 82
ad-hoc tests, 210
adhering to standards, system upgrades, 241
alerting, 197
Application Build and Deployment Document, WebSphere Application Server Build and Configuration Operation, 137
application deployment, Functional and Integration Testing Environment, 142-143
application development and production support teams, 273
feedback, 278-279
respect, 273-274
taking one for the team, 278
teamwork, 274-276
application production support team, 28
application server security, WebSphere Application Server Build Standards Document, 255
application servers, WebSphere Application Server Build Standards Document, 255
application support documentation, production environments, 174
application support teams, 27
striving for operations excellence, 183
applications, tuning, 268
approval from business partners, system upgrades, 240
architects, interproject communication, 297
audits, 36
automation build processes, process engineering, 248
documentation and, 132
server build automation, 253
availability
achieving WebSphere availability, 78-80
severity of problems, 197
system stability and, 294
topology, 130

B
backend systems, stress-testing environment, 154
balanced formal technical training, 70-72
best practices enforcing, Functional and Integration Testing Environment, 144-145
managing fast-changing large projects with engagement, 115-116
bottlenecks, removing, 207
build processes and automation, process engineering, 248
build standards collaboration and review process, process engineering, 246
exception request process, process engineering, 247
process engineering, 244-246
business partner communications and education, WebSphere server builds, 123
business partners, approval for system upgrades, 240
capacity planning, production environments, 179
capacity planning teams, 279-282
case-by-case basis, managing system instability, 284
Category 1 Work, 43
Category 2 Work, 43
centralized technical training, 59-60
change control
  Functional and Integration Testing Environment, 145
  production WebSphere systems stability and stress testing, 169
WebSphere production operations excellence, striving for, 183
change management
  following established processes, 212
  in large interconnected environments, 295-300
change management process documents, Functional and Integration Testing Support, 149
co-production, 314
code defects, fixing, 277
code deployment coordination, production environments, 176-177
collaboration
  enterprise architecture team, 264-265
testing, 268-271
communicating production risks, 163
communication
  following established processes, 211
  interproject communication, improving, 296-297
  production emergency management, 222-224
production WebSphere systems stability and stress testing, 167
servers build, 251
with testing organizations, 270
WebSphere server builds, 123
working with IBM, production emergencies, 229-231
communication failures, production environments, 179
communication skills, hiring experts for WebSphere team, 50-53
communications models, SSME and WebSphere engineering, 316-317
compensation, linking to stability, 290
challenges in implementation, 292-293
incentive plans, 291-292
production-oriented organizations, 293-295
complexity factor, 98
components of WebSphere Operations Framework, 82-84
confidence, lack of (linking stability to compensation), 292-293
configuration change process, Functional and Integration Testing Support, 148
configuration drift, 300
configuration errors, production environments, 177
configuration management, production environments, 177
inconsistencies, 178
configuration standards, WebSphere Application Server, 253
consensus building, post-problem resolution, 219
consistency
large IT projects with multigenerational plans, 35-36
production WebSphere systems stability and stress testing, 168
contractors, hiring experts for WebSphere team, 54-55
converting contractors to employees, 55
coordinators, interproject communication, 297
copying senior management, different perspectives on, 277
credit, sharing with all involved, 275
critical systems, severity of problems, 199
customer experience mitigation, 203
  recycling JVM, 204-205
  system loads, 205-210
customer service failures, severity of problems, 198
data center switch process, WebSphere Data Center Switch Support Operation, 189
data center switch task guide, WebSphere Data Center Switch Support Operation, 189
data collection, troubleshooting, 211
dedicated WebSphere support organization, 8-10
product-based support model, 11-12
WebSphere Planning Engineering, 9
WebSphere Processing Engineering, 9
WebSphere Service Engineering, 10
deep capacity planning, 179
deep dive diagnostic tools, sharing, 183
defects tracking process, Functional and Integration Testing Support, 148
design teams, WebSphere service engineering teams, 20-21
detailed design, 93
development environments, 83
development tooling support, Functional and Integration Testing Environment, 144
distributed platforms, 23
diversity, 262
division of labor, WebSphere system teams, 251-252
DMAIC (define, measure, analyze, improve, control), 99
documentation automation and, 132
following established processes, 211
preparation, post-problem resolution, 219
production environments, 173-175
application support documentation, 174
WebSphere system tracking, 174
documented matrices, severity of problems, determining, 202
documenting production risks, 163
documents, working with IBM (production emergencies), 226-227
dollar amounts, translating WebSphere services into, 104-106
impact on industry, 108
dynamic systems, configuration management (production environments), 177
E education
managing fast-changing large projects with engagement, 113-115
WebSphere server builds, 123
EIA (Enterprise IT Architecture), 263
emergency change-control process, WebSphere Configuration Change Support Operation, 156
enforcing best practices, Functional and Integration Testing Environment, 144-145
engagement, 93
difficulties with, 94-95
IT services, corporate IT culture, 99
IT services, intangible nature of, 95-96
IT services, lack of baseline objectives, 96-97
JVM instances, 97-98
following established processes, 210
three-staged engagement process, 108-110
influence on WebSphere standards, 109
promoting proof of concept, 109
engineering issues with WebSphere server builds, 121-122
engineering processes, managing system instability, 289
engineering support, three-level approach, 27
level 1, 28-29
level 2, 29
level 3, 29-33
engineering support team, Functional and Integration Testing Environment, 140
enterprise architecture team, collaboration, 264
early engagement in design, 264
POC (Proof of Concept), 264
vendor selection, 264
WebSphere strategies, 265
interdependency, 263
Enterprise IT Architecture (EIA), 263
Enterprise Service Bus (ESB), 304
entry-level candidates, hiring for WebSphere team, 55
environment change coordination, 270
Functional and Integration Testing Environment, 146
WebSphere system teams, 252
environment change-coordination team, 13
environmental constraints, testing, 271-272
environments of WebSphere Operations Framework, 82-84
ESB (Enterprise Service Bus), 304
exception request process, build standards and, 247
F facilitators, interproject communication, 296
failover, topology, 130
feedback, work relationships, 278-279
file system layout, WebSphere Application Server Build Standards Document, 255
financial loss, severity of problems, 198
financials, managing during critical situations, 234
Index

fixing code defects, 277
flexibility, WebSphere team, 265
follow up, post-problem resolution, 220-221
form conversion strategies, planning engineering, 243
funding, lack of funding for training, 63
  IBM Education Assistant, 64
  IBM free training, 63
  IBM Proof of Technology programs, 64
  IBM technical expertise, 65
  IBM WebSphere support technical exchange webcasts, 66
  staff meetings, 67

G
global WebSphere workforce, 24
  integrated team models, 24-25
  long-term sourcing strategies, 25-27
globalization, 24
  translating WebSphere services to dollar amounts, 106
guided flexible training, 61-62

H
hiding problems, 200
high level design (HLD), 93, 109
High-Level WebSphere Environment Specifications, 126
hiring WebSphere teams, 42-44, 56
  communication skills, 50-53
  contractors, 54-55
  engagement interviews, 56-57
  entry-level candidates, 55
  interviews, 58
  JEE expertise, 44-45
  proactive hiring, 55
  project management tasks, 45-47
  references, 57
  system expertise, 47-49
  technical interviews, 56
  hiring considerations for WebSphere infrastructures supporting SOA, 308-309
  HLD (high level design), 93, 109
  hosting, 106-107

I
IBM, production emergencies, 225
  communication, 229-231
  documents, generating and uploading, 226-227
  onsite IBM support, 227-229
  relationships, 231-232
IBM Certified SOA Associate, 309
IBM Certified SOA Solution Designer, 309
IBM Certified Solution Developer, 309
IBM Education Assistant, 64
IBM free training, 63
IBM Proof of Technology programs, 64
IBM response, IBM WebSphere Critical Situation, 201
IBM technical expertise, 65
IBM WebSphere Critical Situation, 201-202
IBM WebSphere DataPower System Administration, 308
IBM WebSphere Service Registry and Repository for Administrators, 307
IBM WebSphere support, technical exchange webcasts, 66
improving interproject communication, 296-297
incentive plans
  linking to stability, 291-292
  system stability, 294
inconsistencies, configuration management (production environments), 178
increasing system load capacity, 206-208
industry trends, planning engineering, 244
infrastructure engineering solution team, 12-16
instances, JVM instances (engagement difficulties), 97-98
integrated team models, 24-25
interconnected systems severity of problems, 198
testing, 272
interdependencies, 299
to enterprise architecture team, 263
interproject communication, improving, 296-297
interviews, hiring experts for WebSphere team, 56-58
Introduction to IBM WebSphere Enterprise Service Bus, 307
IT culture
  change in, 316-317
  managing system instability, 289-290
IT organizations complexity and size of, 100-101
  motivation for measuring IT services, 100
IT services, engagement difficulties, 95-96
  corporate IT culture, 99
  lack of baseline objectives, 96-97

J
Java Message Service (JMS), 304
Java Virtual Machine. See JVM
JDK, recycling JVM, 204
JEE
  hiring experts for WebSphere team, 44-45
  WCoE, 37
JEE consulting, Functional and Integration Testing
  Environment, 144
JMS (Java Message Service), 304
JVM (Java Virtual Machine), 97
  recycling, 204-205
  system load capacity, setting changes, 208
JVM (Java Virtual Machine) instances, engagement difficulties, 97-98

L
  large interconnected environments, change management
    organization support, 296-297
    process support, 297-298
    technology support, 298-300
large IT projects with multigenerational plans, 33
  separate WebSphere support, 33-35
  standards and consistency, 35-36
  subteams, building, 33
  leadership teams, severity of problems (determining), 202
level 1 engineering support, 28-29
level 2 engineering support, 29
level 3 engineering support, 29-33
limits of testing, 271
line of business (LOB), 84
  linking stability to compensation, 290
  challenges in implementation, 292-293
  incentive plans, 291-292
production-oriented organizations, 293-295
  listening, building reputations with teams, 236
LLD (low level design), 93, 109
load balancing, topology, 130
LOB (line of business), 8
LOB-based support models, 12, 18
  environment change-coordination team, 13
  infrastructure engineering solution team, 12-16
  separate WebSphere teams, 16-18
long duration tests, 168
production WebSphere systems stability and stress testing, 168
long-term sourcing strategies, 25-27
low level design (LLD), 93, 109

M
  maintenance, WebSphere Application Server Product
    Maintenance Operation, 256-257
  WebSphere Application
    Server System Upgrade
    Check List, 259
  WebSphere Application
    Server System Work Journal, 259
  WebSphere System Work Acceptance Form, 259
managers, peer managers (assisting with production emergencies), 213
managing production emergencies. See production emergency management
  system instability, 284
  actualizing the mean, 287
  dealing with one problem at a time, 285-286
  focusing on system stability, 284-285
  working on what is strategic, 288-290
  teams during critical situations, 232-235
managing fast-changing large projects with engagement, 110-113
  best practices, 115-116
  education, 113-115
  meaning, managing system instability, 287
measuring IT services, motivation for, 100
measuring WebSphere services, recommendations for, 101
managing fast-changing large projects with engagement, 110-116
Service Request, Assignment, and Tracking System, 101, 104-105
three-staged engagement process, 108-110
WebSphere Service Monetization and Benchmark Baseline, 104-108
meetings
  post-problem resolution, 219-220
  progress of production emergency management, 213-214
monitoring, 197
moral boosters, technical training, 72-73
motivation for measuring IT services, 100
multigenerational plans, large IT projects, 33
  separate WebSphere support, 33-35
  standards and consistency, 33-35
subteams, building, 33
naming conventions, WebSphere Application Server Build Standards Document, 254
new initiative resource estimates, 106
no hardware growth strategies, 312

objectives of WebSphere Operations Framework, 81-82
offshorting. See global WebSphere workforce onsite IBM support, working with IBM (production emergencies), 227-229
operating system security, WebSphere Application Server Build Standards Document, 255
operation specification, 82
operations, 82
WebSphere Functional and Integration Testing Support, 146-148
change management process document, 149
configuration change process, 148
defects tracking process, 148
weekly testing reports, 149
operations attributes, WebSphere Operations Framework, 90-91
optimizing applications, testing, 268
organization support, change management in large interconnected environments, 296-297
organizational models, SSME and WebSphere engineering, 315-316
organizations, managing system instability, 288-289

Pay for Performance (PFP), 108 peer managers, assisting with production emergencies, 213
performance topology, 130
WCoE, 38-39
perspectives on copying senior management, 277
developing ability to look at issues from different perspectives, 276-278
on fixing code defects, 277
taking one for the team when problems arise, 278
on WebSphere configuration, 277
PFP (Pay for Performance), 108
pipelines, Functional and Integration Testing Environment, 141
planning tests, 270
planning engineering form conversion strategies, 243
industry trends, 244
system upgrade strategies, 243-244
POC (proof of concept), 109
collaboration, enterprise architecture teams, 264
Post-Change Peer Review Document, WebSphere Configuration Change Support Operation, 156
post-problem resolution, 218
follow up, 220-221
meetings, 219-220
preparation consensus building, 219
documentation, 219

post-problem resolution, 218
pre-production environments, 84
priorities, collaborating with testing organizations, 269
private messages, sharing credit for job well done, 275
problem evaluation and reporting, severity of problems, 199
problem reporting following established processes, 211
severity of problems, determining, 203
problem resolution, WebSphere Application Server Problem Resolution Support Operation, 218
Problem Resolution Activity Diagram Document, 218
Problem Resolution Check List, 218
problem reviews distracting senior executives, 287
managing instability, 285
stressing the whole organization, 286
problem statements, severity of problems, determining, 203
problems hiding, 200
from insufficient stress testing, 161-163
over reporting, 201
responsibility, 237
under reporting, 199-200
process consistency, process engineering, 245-246
Index

process engineering, 244
build process and automation, 248
build standards, 244
collaboration and review process, 246
and exception request process, 247
and process consistency, 245-246
standard and process team, 245
process support, change management, 297
in large interconnected environments, 297-298
processes, progress of production emergency management, 210
change management, 212
communication and teamwork, 211
documentation, 211
engagement, 210
problem reporting, 211
testing, 212
troubleshooting data collection and upload, 211
product-based support model, 8
dedicated WebSphere organization, 11-12
product-specific training for engineers who build WebSphere infrastructures supporting SOA, 307-308
product support responsibilities, WebSphere infrastructure technologies with SOA, 308-309
production emergencies, assisting peer managers, 213
production emergency management, 196-197
best practices
documented matrices, 202
leadership teams, 202-203
problem statements, 203
communication, 222-224
customer experience mitigation, 203
recycling JVM, 204-205
system loads, 205-210
IBM, communication, 229-231
documents, generating and uploading, 226-227
onsite support, 227-229
relationship with, 231-232
working with, 225
invoking IBM WebSphere Critical Situation, 201-202
managing progress, 210
following established processes, 210-212
meetings and status reporting, 213-214
senior management, 212-213
over reporting problems, 201
post-problem resolution, 218
follow up, 220-221
meetings, 219-220
preparation, 218-219
severity, 197
availability, 197
critical systems, 199
customer service failures, 198
financial loss, 198
interconnected systems, 198
unstructured problem evaluation and reporting, 199
team management during critical situations, 232-235
teamwork, 224-225
under reporting problems, 199-200
WebSphere Application Server Problem Resolution Support Operation, 215-217
problem resolution process, 218
production environment support
capacity planning, 179
code-deployment coordination, 176-177
communication failures, 179
configuration management issues, 177
inconsistencies, 178
documentation, lack of, 173-175
application support documentation, 174
WebSphere system tracking, 174
testing, lack of, 175-176
tooling, lack of, 178
upgrade strategy challenges, 180
production environments, 171-172
WebSphere Application Release Support Operation, 184-186
WebSphere application release form, 187
WebSphere application release management process, 186
WebSphere system configuration comparison process, 186
WebSphere Data Center Switch Support Operation, 187-189
data center switch process, 189
data center switch task guide, 189
WebSphere configuration comparison program, 190
production environments, 84
future of, 192-193
production failures, recreating, 268
production operations, striving for excellence in, 181
application support teams, 183
change control, 183
disciplined and technical teams, building, 181-182
production system documentation, 182
sharing deep dive diagnostic tools, 183
production risks, communicating, 163
production support teams, 27
production system documentation, 182
production WebSphere systems, stability and stress testing, 167
change control, 169
communication, 167
consistency, 168
long duration tests, 168
production-oriented organizations, 293-295
progress, production emergency management, 210
following established processes, 210-212
meetings and status reporting, 213-214
senior management, 212-213
project management, hiring contractors, 54
experts for WebSphere team, 45-47
project planning, 46
project resource estimates, 46
Proof of Concept (POC) collaboration, enterprise architecture team, 264
three-staged engagement process, 109
quantitatively measuring WebSphere services, 101
Service Request, Assignment, and Tracking System, 101, 104-105
WebSphere Service Monetization and Benchmark Baseline, 104-108
recognizing teams for jobs well done, 276
recommendations for measuring WebSphere services, 101
managing fast-changing large projects with engagement, 110-116
Service Request, Assignment, and Tracking System, 101, 104-105
three-staged engagement process, 108-110
WebSphere Service Monetization and Benchmark Baseline, 104-108
recreating production failures, 268
recycling JVM, customer mitigation, 204-205
reducing system loads, 205
references, hiring experts for WebSphere team, 57
regenerating WebSphere engineering, 313-314
regression tests, 209
relationships, 261. See also work relationships
working with IBM, production emergencies, 231-232
release management, process support, 298
removing system bottlenecks, 207
reports, weekly testing reports (Functional and Integration Testing Support), 149
reputations building with WebSphere teams, 235-238
WebSphere teams, 206
Request for Proposal (RFP), 264
resource consumption, IBM WebSphere Critical Situation, 201
resources, Functional and Integration Testing Support, 149-150
respect, work relationships, 273-274
responsibilities, taking responsibilities of problems, 237
review process, build standards (process engineering), 246
RFP (Request for Proposal), 264
rotation, team stability and, 69
routing traffic away from problem components, system load capacity, 209-210
scheduling system upgrades, 241
upgrades, WebSphere system teams, 250
Index

security
application server security, WebSphere Application Server Build Standards Document, 255
operating system security, WebSphere Application Server Build Standards Document, 255
topology, 131
WebSphere system
topology, 131
teams, 253
senior management
copying, 277
progress of production emergency management, 212-213
senior management review, process support, 298
server build automation, WebSphere system teams, 253
server build
communications, 251
server consolidations, 310
Service Request, Assignment, and Tracking System,
measuring WebSphere services, 101-105
Service Science, Management and Engineering. See SSME services systems, SSME and WebSphere engineering, 317-318
severity of problems
best practices for determining documented matrices, 202
leadership teams, 202-203
problem statements, 203
production emergency management, 197
availability, 197
critical systems, 199
customer service failures, 198
financial loss, 198
interconnected systems, 198
unstructured problem evaluation and reporting, 199
shared WebSphere computing environments, 312
sharing credit for job well done with all involved, 275
sharing deep dive diagnostic tools, 183
shutting down problem components, system load capacity, 208-209
SME (Subject Matter Expert), 110
SOA, WebSphere infrastructure technologies, 304
product support responsibilities and hiring considerations, 308-309
skill sets and training considerations for, 305-308
software incompatibility, WebSphere server builds, 122
SSME (Service Science, Management and Engineering), 303
WebSphere engineering and, 314-315
communications model, 316-317
engineering processes, 315
organizational model, 315-316
services systems, 317-318
stability
achieving WebSphere stability, 78-80
linking to compensation, 290
challenges in implementation, 292-293
incentive plans, 291-292
production-oriented organizations, 293-295
production WebSphere systems and stress testing, 167
change control, 169
communication, 167
consistency, 168
long duration tests, 168
teams, 69
staff meetings, training, 67
standard and process team, process engineering, 245
standards
adhering to for system upgrades, 241
build standards, process engineering, 244-247
large IT projects with multigenerational plans, 35-36
three-staged engagement process, 109
status reporting, progress of production emergency management, 213-214
stress, teams and, 233
stress testing, 154
future of, 169-170
problems resulting from insufficient testing, 161-163
production environments, 175
production WebSphere systems stability and, 167
change control, 169
communication, 167
consistency, 168
long duration tests, 168
stress-testing environment support, 153-154
challenges for, 164
interconnected systems and increased workload and complexity, 164-165
resource intensive for WebSphere team, 165-167
stress-testing environments may not be built to perform stress testing on interconnected systems, 165
WebSphere Configuration Change Support Operation, 155-156
emergency change-control process, 156
Post-Change Peer Review Document, 156
WebSphere System Tracking Document, 157
WebSphere Stress-Testing Support Operation, 157-160
Problem Resolution Guideline and Procedure, 160
stress-testing environments, 83, 269
WebSphere engineering support life cycles, 88-89
stress tests, 209
Subject Matter Expert (SME), 110
subteams, building, 33
Supervisory Call (SVC), 204
support for Functional and Integration Testing Environment, 140
application deployment, 142-143
change control, 145
development tooling support, 144
enforcing WebSphere and JEE best practices, 144-145
environment change coordination, 146
JEE consulting, 144
support models, 140-141
system consistency and upgrade strategies, 142
testing pipelines, 141
third-party software, 141-142
WebSphere Network Deployment, 145
WebSphere Virtual Enterprise, 145
support models, Functional and Integration Testing Environment, 140-141
supporting testing organizations, 272
SVC (Supervisory Call), 204
SVC dumps, 205
system bottlenecks, removing, 207
system consistency, Functional and Integration Testing Environment, 142
system experts, hiring experts for WebSphere team, 47-49
system instability, managing, 283-284
actualizing the mean, 287
dealing with one problem at a time, 285-286
focusing on system stability, 284-285
working on what is strategic, 288-290
system loads, customer mitigation, 205
increasing capacity, 206-208
JVM setting changes, 208
reducing, 205
routing traffic away from problem components, 209-210
shutting down problem components, 208-209
system stability availability and, 294
focusing on, 284-285
incentive plans, 294
system teams, WebSphere service engineering teams, 21-22
system testing environments, 83
system tests, 209
system upgrade strategies, planning engineering, 243-244
system upgrades, WebSphere Application Server adhering to standards, 241
approval from business partners, 240
necessity of, 242
scheduling, 241
system visualization, 299
TCP/IP port assignments, WebSphere Application Server Build Standards Document, 255
team management during critical situations, 232-235
team system architect, 35
teams, 85
application development and production support teams, 273
develop ability to look at issues from different perspectives, 276-278
feedback, 278
respect, 273-274
taking one for the team, 278
teamwork, 274-276
application support teams, 27
striving for operations excellence, 183
capacity planning teams, 279-282
enterprise architecture team, 263-265
collaboration, 264-265
interdependency, 263
environment change-coordination team, 13
infrastructure engineering solution team, 12-16
integrated team models, 24-25
leadership teams, severity of problems, 202
long-term sourcing strategies, 25-27
recognition for, 276
separate WebSphere teams, 16-18
standard and process teams, process engineering, 245
stress, 233
subteams, building, 33
WebSphere production operations excellence, striving for, 181
application support teams, 183
change control, 183
disciplined and technical teams, building, 181-182
production system documentation, 182
sharing deep dive diagnostic tools, 183
WebSphere service engineering teams, 20
dedicated environments on distributed platforms, 22-23
design team, 20-21
system team, 21-22
WebSphere system teams, 248-250
configuration standards, 253
division of labor, 251-252
environment change coordination, 252
security, 253
server build automation, 253
tracking large environments, 252
upgrade scheduling and coordination, 250
WebSphere and Java Best Practices Document, 254
WebSphere teams flexibility, 265
lessons learned from managing, 77-78
teamwork following established processes, 211
production emergency management, 224-225
work relationships, 274-276
technical environment change coordination, WebSphere Application Server and Web Server Build and Configuration Operation, 138
technical training building strong teams, 58
centralized technical training, 59-60
decentralized technical training, 60-61
guided flexible training, 61-62
lack of funding, 63-67
inclusive training, 270
as morale booster, 72-73
technology support, change management in large interconnected environments, 298-300
configuration drift, 300
terminology of WebSphere Operations Framework, 82
testing, 266-267
communication, 176
coordination, 267-268
environmental constraints, 271-272
following established processes, 212
knowing limits of, 271
planning tests, 270
production environments, 175-176
testing meetings, 214
testing organizations, 266-267, 273
collaboration, 268-271
communication with, 270
coordination, 267-268
supporting, 272
testing strategies, 269
tests ad-hoc tests, 210
regression tests, 209
stress tests, 209
system tests, 209
third-party software, Functional and Integration Testing Environment, 141-142
three-level approach to engineering support, 27
level 1, 28-29
level 2, 29
level 3, 29-33
three-staged engagement process, 108-110
influence on WebSphere standards, 109
promoting proof of concept, 109
tooling, production environments, 178
tools, deep dive diagnostic tools, 183
topology, WebSphere Application Server and Web Server Topology Design Operation, 127-128
WebSphere Operation Framework and Description, 128-130
WebSphere Topology Design Checklist, 130-131
Copyright © 2006 IBM Corporation. All rights reserved.
WebSphere Topology Design Diagram, 131
WebSphere Application Server build, 87, 251
WebSphere Application Server Build Standards Document, 254-255
WebSphere Application Server Monitoring Requirements Document, 191
WebSphere Application Server Planning Engineering form conversion strategies, 243 industry trends, 244 system upgrade strategies, 243-244
WebSphere Application Server Problem Resolution Support Operation, 215-217 problem resolution, 218
WebSphere Application Server Process Engineering, 244 build process and automation, 248 build standards, 244 and exception request process, 247 and process consistency, 245-246 collaboration and review process, 246 standard and process team, 245
WebSphere Application Server Product Maintenance Operation, 256-257
WebSphere Application Server System Upgrade Check List, 259
WebSphere Application Server System Work Journal, 259
WebSphere System Work Acceptance Form, 259
WebSphere Application Server System Tracking Document, 252
WebSphere Application Server System Upgrade Check List, 259
WebSphere Application Server System Work Journal, 259
WebSphere Application Support Document, WebSphere Application Server and Web Server Build and Configuration Operation, 137
WebSphere Build Request Document, WebSphere Application Server and Web Server Build and Configuration Operation, 136
WebSphere Center of Excellence (WCoE), 36-39 JEE, 37 performance testing and monitoring, 38-39
WebSphere configuration, perspectives on, 277
WebSphere Configuration Automation Process, WebSphere Application Server and Web Server Build and Configuration Operation, 136
WebSphere Configuration Change Support Operation, 154-156 emergency change-control process, 156
Post-Change Peer Review Document, 156
WebSphere System Tracking Document, 157
WebSphere configuration comparison program, WebSphere Data Center Switch Support Operation, 190
WebSphere Data Center Switch Support Operation, 187-189 data center switch process, 189
data center switch task guide, 189
WebSphere configuration comparison program, 190
WebSphere DataPower, 308
WebSphere Engineering, 76 achieving stability and availability, 78-80 defined, 77 lessons learned from managing WebSphere teams, 77-78 regenerating, 313-314 SSME and, 314-315 communications models, 316-317 engineering processes, 315 organizational models, 315-316 services systems, 317-318
WebSphere engineering support life cycles, 82, 86 application migration and deployment, 89 production support, 89 stress-testing environments, 88-89 upgrades, 89
WebSphere Application Server Build, 87
WebSphere Functional and Integration Testing, 88
WebSphere Initial Sizing and Consulting Engagement phase, 86
WebSphere engineering task classification, 101-102 WebSphere engineering teams, 19-20 virtual teams, 19
WebSphere service engineering teams, 20 dedicated environments on distributed platforms, 22-23
<table>
<thead>
<tr>
<th>WebSphere Resource Estimation and Hosting Charge Metrics Document, 127</th>
<th>WebSphere System Monitoring Support Operation, 190-191</th>
<th>decentralized technical training, 60-61</th>
</tr>
</thead>
<tbody>
<tr>
<td>WebSphere server builds business partner communications and education, 123 engineering issues, 121-122</td>
<td>WebSphere Application Server Monitoring Requirements Document, 191</td>
<td>lack of funding, 63-67</td>
</tr>
<tr>
<td>WebSphere Service Engineering, 10, 43</td>
<td>WebSphere system teams, 248-250 configuration standards, 253 division of labor, 251-252 environment change coordination, 252 scheduling upgrades, 250 security, 253 server build automation, 253 tracking large environments, 252</td>
<td>flexibility, 265</td>
</tr>
<tr>
<td>WebSphere service engineering teams, 20 dedicated environments on distributed platforms, 22-23 design team, 20-22</td>
<td>WebSphere and Java Best Practices Document, 254</td>
<td>hiring, 42-44, 56 communication skills, 50-53 contractors, 54-55 engagement interviews, 56-57 entry-level candidates, 55 interviews, 58 JEE expertise, 44-45 proactive hiring, 55 project management tasks, 45-47 references, 57 system expertise, 47-49 technical interviews, 56 lessons learned from managing, 77-78 rotation and stability, 69 training balanced formal technical training, 70-72 technical training, 72-73</td>
</tr>
<tr>
<td>WebSphere Service Monetization and Benchmark Baseline, 104-107 impact on industry, 108</td>
<td>WebSphere system tracking, documentation (production environments), 174</td>
<td>WebSphere engineers, 69</td>
</tr>
<tr>
<td>WebSphere System Tracking Document, WebSphere Configuration Change Support Operation, 157</td>
<td>WebSphere System Work Acceptance Form, 259</td>
<td>WebSphere topology, 127</td>
</tr>
<tr>
<td>WebSphere teams, 8, 85 building reputations with, 235-238 building strong teams by focusing on technical training, 58 centralized technical training, 59-60</td>
<td>WebSphere System Build Acceptance Form, 252</td>
<td>WebSphere Topology Design Checklist, 130-131</td>
</tr>
<tr>
<td>WebSphere Virtual Enterprise, 311 Functional and Integration Testing Environment, 145, 150 optimizing, 311-313 stress testing, 170 weekly testing reports, Functional and Integration Testing Support, 149</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
weight considerations for
WebSphere services, 103-104
work relationships, 261-262
application development
and production support
teams, 273
feedback, 278-279
respect, 273-274
taking one for the
team, 278
teamwork, 274-276
capacity planning teams,
279-282
taking one for the
team, 278
teamwork, 274-276
capacity planning teams,
279-282
enterprise architecture team,
263-265
collaboration, 264-265
interdependency, 263
testing organizations,
266-267, 273
collaboration, 268-271
coordination, 267-268
WebSphere teams,
flexibility, 265
workload considerations,
WebSphere Functional and
Integration Testing Support,
149-150
WSRR (WebSphere Service
Registry Repository), 304, 308

X-Y-Z
XML, WebSphere infrastructures
supporting SOA, 305-306