N1 Grid Architecture Realized: Measurable Requirements

Jason Carolan, Sun Client Services
Scott Radeztsky, Sun Client Services
Paul Strong, Software Marketing
Ed Turner, Sun Client Services

Sun BluePrints™ OnLine—March 2005

http://www.sun.com/blueprints

Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95045 U.S.A.
(650) 960-1300

Part No. 819-1742-10
Revision A, 2/28/05
Edition: March 2005

Please
Recycle

Adobe PostScript™

N1 Grid Architecture Realized: Measurable Requirements

Note – This article is the fifth chapter of the Sun BluePrints™ book *Building N1 Grid Solutions: Preparing, Architecting, and Implementing Service-Centric Data Centers* by Jason Carolan, Scott Radeztsky, Paul Strong, and Ed Turner, which will be available through `www.sun.com/books`, `amazon.com`, and Barnes & Noble bookstores.

This chapter discusses using the Sun architecture methodologies to translate customer business drivers and stated functional and operational requirements into a measurable Critical to Quality (CTQ) baseline for architectural analysis and solution testing. This chapter discusses techniques for collecting and analyzing CTQ baselines that are created with clearly traceable origins, clearly measurable success criteria, and clearly documented activities to make those CTQ measurements. This chapter also describes how the N1 Grid architecture can enable implementors to think of measuring and delivering a very different set of CTQs than previously available to them and some of the work required to collect, deliver, and use those measurements.

Two types of measurements are discussed in this chapter—those that support the architecture methodology phases and those that serve as baselines for new CTQs. The measurements that support the architecture methodology are used to inform the architectural analysis and verification phases and enable the use of data-driven decisions at a tollgate where advancement to the next phase of the architecture methodology is being considered. The measurements that serve as baselines for new CTQs often require up-front preparation before that CTQ data can be measured or leveraged because IT organizations have not thought of pulling together the people, processes, and tools to measure the new CTQs that the N1 Grid solutions enable. IT organizations typically do not have this data collected and reported in data centers because the new N1 Grid CTQs are more business oriented.

Two examples that many businesses do not currently measure are:

- Average time to market for a new business service. Businesses using the N1 Grid system reduce the time it takes to move a service through its production life cycle.
- Business services released during the year. By leveraging reduced production service life cycle times, businesses can count on that efficiency and release more services per year.

This chapter opens by presenting techniques that sharpen requirements gathering skills by focusing them on areas that N1 Grid technology emphasizes—business drivers, architecture, and measurements that enable data-driven decision making. This chapter provides an overview of use cases and service life cycles that Sun and Sun customers have found extremely useful to guide documentation of the N1 Grid software functional and operational activities and capture derived requirements. These use cases and requirements serve as the basis for development and preproduction verification testing, as well as ongoing CTQ reporting in the final production environment. This chapter then discusses traditional CTQs and new N1 Grid CTQs on which requirements gathering can focus, along with examples to help you think about the new N1 Grid business solution space. The chapter closes with ideas to help you measure and discuss operational readiness and change acceptance, and it prepares you for the activities in the next chapter—the architecture analysis phase.

Traceable Requirements

Requirements are an output of architecture because N1 Grid requirements are derived as part of the process of getting to the architecture, rather than trying to design the architecture from raw statements of need from stakeholders. The raw Voice of the Customer (VOC) statements serve as a basis for a series of activities, such as:

- Requirements gathering
- Requirements analysis
- Requirements validation and verification

All three activities are designed to help organize and prioritize requirements.

Requirements serve as a link between key business drivers and the architecture, people, processes, and products that make up the solution. Requirements should establish the measurable success criteria for the solution in terms of the operational, functional, and service-level objectives. The gathering of requirements is an iterative process. It must start with an identification of stakeholders and other places from which requirements might be gathered. The stakeholders can be the source of explicit raw VOC requirements, as well as the validators of activities that identify implied requirements and an understanding of the constraints and assumptions across the business, functional, and operational boundaries.

As shown in FIGURE 0-1, iteration occurs when stakeholders are both the source and validators of business drivers, use cases, service level requirements (SLRs), and information derived from architectural or operational assessments. It is important to capture the ownership of the requirement (that is, the person who will say that the requirement has been met) and a means for requirement acceptance (an agreement for how the requirement is to be satisfied) at the same time the requirement is documented.

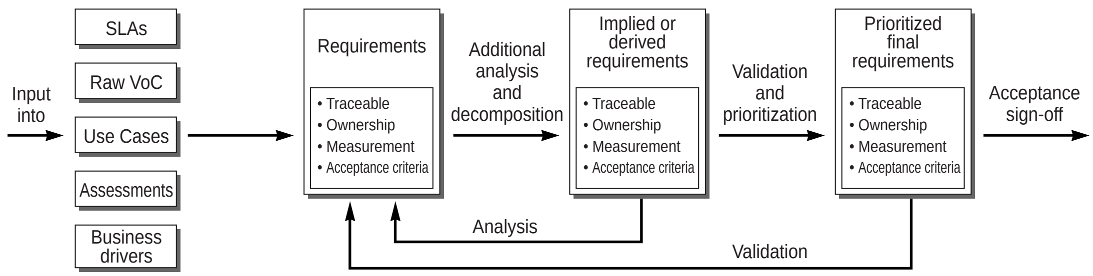

FIGURE 0-1 Stakeholder Activities During the Requirements Phase

The boxes represent possible artifacts of the requirements process where stakeholder information is documented. The arrows show how stakeholders can be active in all phases of the requirements gathering, analysis, validation, and approval phases.

In addition to their activities in the requirements gathering phase, the stakeholders and requirements have interaction throughout the entire architecture life cycle, as shown in FIGURE 0-2.

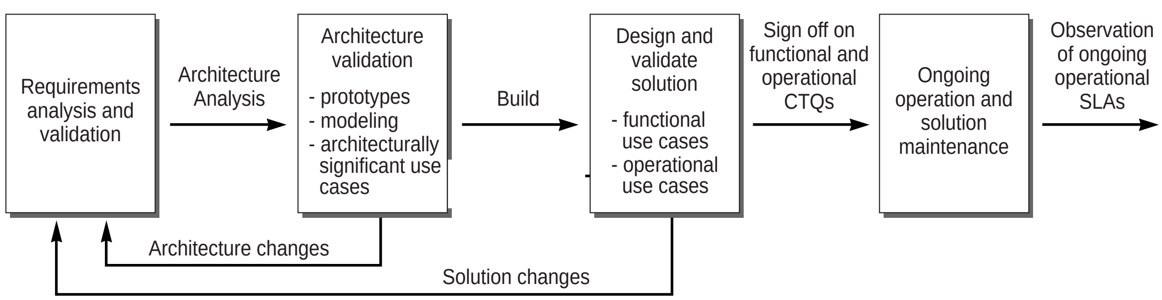

FIGURE 0-2 Stakeholder and Requirements Interaction

This diagram shows that the requirements can be updated throughout the architectural process as new information is uncovered. Although not drawn in FIGURE 0-2, these iterative updates might also eminate from the ongoing operation and solution maintenance box. Experiences with complex network computing problems bear this out:

- Prototypes and other side projects that quickly model some aspect of the architecture can uncover previously unknown opportunities or solidify previously undefinable requirements. This is usually the exact purpose of inserting them into the architecture analysis process.
- Requirements contribute to the architectural analysis by guiding the choices that are made and way they are presented for design and validation (for example, there are many ways to deliver single sign-on, and some might be in conflict with other functional or operational requirements).
- Requirements contribute to the validation of the design phase by documenting the testing of the functional (for example, “This must happen when I click this button,” and “The architecture must support 10,000 simultaneous clicks.”) or the operational (for example, the people and processes are in place to respond to this alert) use cases required to demonstrate a working solution.
- Requirements contribute to the ongoing operation of the solution (for example, the method exists to update parts of the solution without service interruption) and guide the mechanisms to collect and places to report information that demonstrate the delivery of SLRs captured in the requirements artifacts.

Traceable requirements are the key to enable the prioritization, validation, and measurement of the requirements activities just discussed. Traceability of requirements has several implications:

- All project activities can be linked to the analysis, delivery, measurement, or reporting of a captured requirement.
- All captured requirements can be traced back to key business drivers (KBDs).
- All captured requirements can be directly measured or decomposed into directly measurable derived requirements.
- All new information from the iterative architecture methodology can be propagated to the appropriate existing requirements for deprecation or updating
- All measurement techniques for changed, updated, or deprecated requirements can be likewise changed, updated, or deprecated
- All captured requirements have an owner who could sign off after the demonstrated delivery of that requirement.

FIGURE 0-3 demonstrates this traceability for a fictitious subset of requirements.

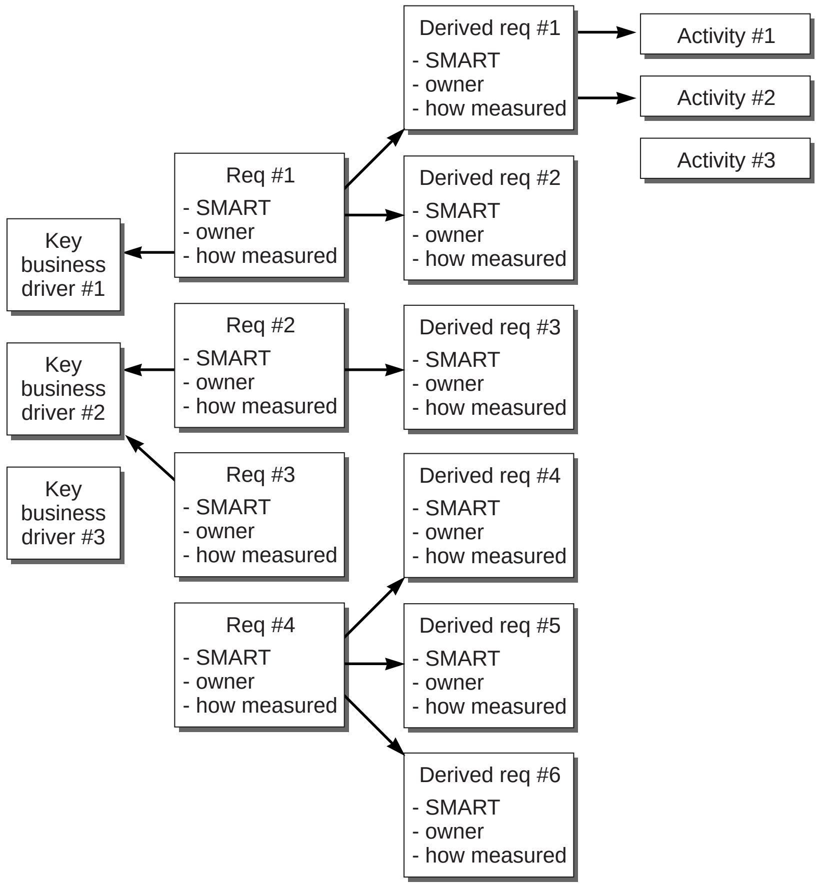

FIGURE 0-3 Requirements Traceability

For discussion, suppose that the items in FIGURE 0-3 are the current state of the documented requirements. The following discussion is based on the just discussed bullet points:

- All of the requirements should have an owner. Owners serve as the source of initially derived requirements and of information that determines how the requirement has been delivered. In this way, requirement owners provide information, but they are also accountable for the process to verify the delivery of the features they demand.
- All of the requirements should point to the KBDs they support. In the diagram, the fourth requirement has been captured, and although it has been decomposed into derived requirements, it does not point back to a KBD. Such items are either points of interest (that is, they illustrate an important missing KBD or requirement), or they might turn out to be items on someone's agenda collected during the gathering

process, but not aligned with the rest of the project requirements. In addition, the third KBD does not currently have requirements supporting its existence. This fact should either motivate its removal or point out that additional analysis must be undertaken to uncover stakeholders and information that support it.

- All of the requirements should be directly measurable or eventually decomposed into directly measurable derived requirements. In this case, the owner of the third requirement must agree that it is completely measurable as captured; otherwise, the requirement should be rejected or additional decomposition should occur. The fourth requirement was decomposed into multiple derived requirements. Each derived requirement (four through six) must be measurable in a way that is acceptable to its owner, and the derived requirement measurements combined in a way that is acceptable to the owner of the fourth requirement.
- All of the project activities on the far right should be linked to the requirement they address. The third activity is an example of project activity that should probably be challenged. It has somehow been identified as needed, but it has not yet been linked to a requirement it supports in terms of adding value.

The iterative nature of analysis and traceability fills out the links and documents the priorities (that is, prioritized KBDs). In this way, you do not need to work solely from left to right. You can start with any mix of drivers, activities, and requirements and still end up with a completed list of traceable requirements. The requirements analysis phase is often where many of the derived requirements and implied constraints emerge. They are added into the table, as solutions are worked out more completely in pilots and benchmarks and illuminate or disprove some functionality.

The validation and verification activities finalize and socialize the requirements and CTQs, ensuring that the stakeholders agree and sign off to the requirements as a complete and accurate description of the desired future state.

SMART Requirements

This section discusses how to make your measurements more useful. Although the definition for “T” is a variation on the acronym’s traditional definition, it is important to capture requirements that are SMART:

- Specific – The description and purpose is clear and unambiguous.
- Measurable – The requirement is quantifiable, observable, verifiable, and testable.
- Attainable – The requirement is technically or operationally possible (for instance, 101% uptime is not).

- Realizable – The requirement is possible to implement in the current context of the organization (for example, 50% uptime may not be realizable in a datacenter with no roof).
- Traceable – The requirement can be linked back to the KBDs and linked forward to requirement validation and acceptance criteria activities, as discussed in the previous section.

If a particular requirement is not SMART, it needs to be broken down into additional requirements and constraints that are SMART. The acceptance criteria for each requirement should be gathered at the same time as the requirement. By using the collected methods, the tests performed or qualities demonstrated in the final solution should prove that the requirements have been satisfied in a completely unambiguous manner.

Poorly formed service level requirements (SLRs) lead to ambiguity and unmeasurable or unverifiable success criteria. Properly formed SLRs should at least contain the following information:

- Stakeholder
- Goal
- Context
- Range
- Hazard response

The stakeholder is the person or organization who owns or is driving the SLR. The stakeholder will need to see the output of your SLR verification. The presence of this SLR means that the stakeholder has already stated the goal, so it is your starting point for negotiation or analysis regarding how to measure or demonstrate that the goal has been, or will be, met.

The context should include the use cases or scenarios in which this goal is essential. The context sets some constraints on the SLR by describing the environment and exact steps the actor performs when the SLR will be measured or verified. In this way, the context provides some level of an unambiguous environment in which the SLA measurement will be made.

The range begins to outline possible acceptable outcomes, if appropriate. Are they really asking for exactly a 5-millisecond response time? Or, is a 2-millisecond to 7-millisecond response time acceptable? Is 80 percent less than 5 milliseconds another possible way to categorize this goal? These considerations can have important influences on the a solution cost and design choices.

The hazard response outlines what should happen when the SLR is not met (for example, when transaction time is above the threshold). For every SLR, there are three types of hazard responses to consider and discuss: measures you take to prevent the problem from happening in the first place, measures taken to alert someone before the problem goes too far, and a contingency plan for what to do when the problem does occur. The hazard responses that a customer is willing to build into the architecture are an indication of the importance of the requirement. Because hazard responses often involve people and process, they are also a good source of derived operational requirements.

Use Cases

The classic definition of a use case is that it describes a sequence of actions that are performed by a system to yield a result of value to a user. In simple terms, use cases describe what the actors do and what they want the system to do. Use cases present another opportunity for measurement and act as a source of activity that can be measured. The previous sections mentioned use cases as a potential basis for both derived and explicit requirements and constraints, as well as a means of establishing the context for a measurable SLR.

The sequence referred to in the definition is a specific flow of events through the system or an instance. Identifying and describing your use cases means identifying and describing a group of related flows of events. Actions are computational or algorithmic procedures, either invoked when the actor provides a signal to the system or when the system gets a time event. An action can imply signal transmissions to either the invoking actor or other actors. An action is atomic, which means it is performed either entirely or not at all. You can put a value on a successfully performed use case. This is very important to determine the correct level of granularity for a use case, to ensure that you are not achieving use cases that are either too minute to be useful or too large in number for the project's scale. In the unified modelling language (UML), actors are external to the system and always start off the use case.

The N1 Grid vision is about unleashing organizational potential, so your N1 Grid strategy tasks should be worked out into specific use cases that will help you illuminate, prepare for, and prove that the people, process, and technology choices you made deliver the solution you are designing.

Some functional use case examples might be:

- Users are properly authenticated and able to access and view properly formatted information from wireless PDA.
- Ten thousand properly authenticated users are simultaneously able to access and view properly formatted information from ten thousand wireless PDAs.
- Systems are properly correlating multiple event and information data points every ten minutes for business SLR reporting.

- Users can browse the repository of automated N1 Grid software installation plans for Enterprise JavaBeans deployment.
- Users can choose an N1 Grid software installation plan for Enterprise JavaBeans based on a business policy for deployment into the production environment.

Operational use case examples to test might include:

- Users are updating Enterprise JavaBeans in application server containers as part of the test-to-production service life cycle update.
- Users or the system are moving additional applications onto compute resources according to a business policy in response to an observation of spare CPU capacity.
- Help desk is appropriately responding to an alert in the management console according to problem management and escalation procedures.
- Users are using the N1 Grid software to quickly reconfigure the test environment to its exact configuration on midnight of June 7 of last year in response to a regulatory demand.

CTQ Measurements

People are often not prepared for the work required to collect, roll up, analyze, and report CTQs—much less for the effort to preemptively react to maintain CTQ levels against events or other impacting situations. Beyond these common gaps, IT has typically not been run with business-level CTQs, nor has it worked out relationships between system-level events and information from probes and agents to components that help a business unit to make business decisions. The IT organization has been in the business of keeping the disks spinning and the servers running, and the CTQs for those types of activities are often granular enough to not require analysis at a higher level than disk metering and an occasional `ping`(1M) command to the server. Integration is occasionally done, but only for a few combinations of components. Typical questions are:

- How is the disk stripe doing in the array?
- Was there excessive network traffic at a specific time (for instance, during a denial-of-service attack that finally brought down some of the servers)?

The ability of the N1 Grid software to enable operational agility further up the stack also presents an opportunity to observe and use information from the layers further up the stack—layers more closely connected to the business applications and services. Although people and processes must be in place to use this information after it is collected, the basis of observation and the information that can be used should come from an organized framework that includes all of the needed layers. The next section illustrates a framework that can organize the collection and correlation of business information.

OMCM Tool Framework

Chapter 4 introduced the Operational Management Capabilities Model (OMCM) as a means to help define key process areas that provide a measurement of the people, processes, and tools in a given organization. For visibility into the enterprise stack, the tools component of the OMCM details the functional components and their relationships to enterprise management technology. The tools are used to measure and report the performance of the system and to provide visibility into and reporting of the operational systemic qualities of the system.

The management tools framework (FIGURE 0-4) consists of a layered combination of management applications that are tied together, when appropriate, through the integration of specific components. This section contains a brief description of the management tools framework. For more information, refer to the Sun white paper "Operational Management Capabilities Model" by Michael Moore and Edward Wustenhoff.

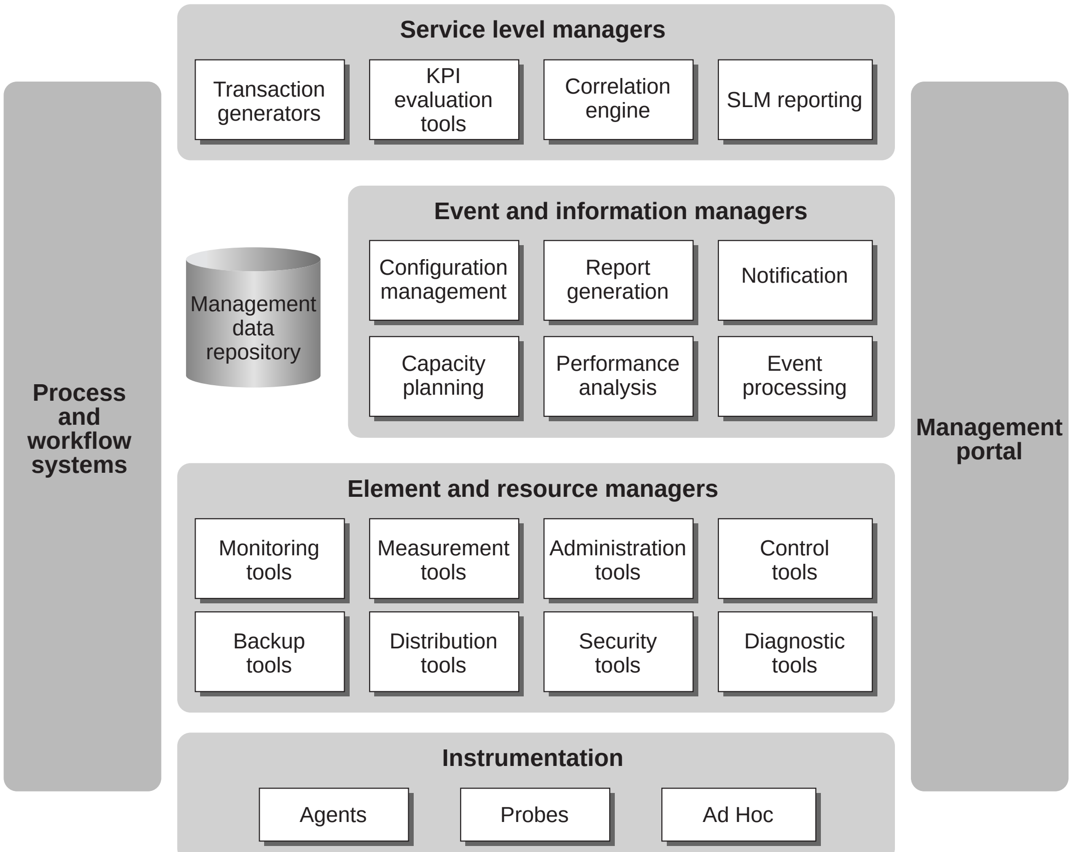

FIGURE 0-4 Management Tools Framework

- The *instrumentation layer* consists of all management elements that enable the various management tools to gain access to managed resources. Instrumentation is generally implemented within the context of the execution framework where managed resources reside through the appropriate agents, probes, or other adhoc scripts and executables.
- The *element and resource management layer* consists of management applications that directly interact with the execution environment to query or modify managed resources.
- The *event and information layer* consists of applications that manage events and information generated by the lower layers of the framework. The focus of the applications at this layer shifts from the measurement and modification of technical metrics to the management of data and alarms.

- The *management data repository* is a logical representative of the storage and management of operational data.
- *Service-level managers* are applications that provide the tie-in between business requirements as defined by SLAs and the technical status of the execution environment as determined by the lower layers of the framework.

Workflow technology is used to automate the management processes described in the IT Management Framework (ITMF). Examples of this type of technology are a trouble ticket system to support the problem management process or the automation of a change approval system used to support the change management process.

The management portal is a collection of applications that provides external entities access to selected portions of the management framework. Examples would be a web interface for reviewing service level management (SLM) reports, web or other types of user interfaces for various tools, or an application used by end users to submit requests for a service. It should also be possible, and even desirable, to use this portal to expose management information and facilities to people outside of the IT organization.

Example CTQs

The N1 Grid architecture enables businesses to consider new types of activities gauged by new types of CTQs. This section provides an overview of some of those business CTQs and a discussion of the ways that N1 Grid solutions can begin to help you focus on and deliver value to the business represented in those CTQs.

New Systems

In this book's Foreword, Greg Papadopolous emphasized the changes and opportunities coming to business organizations—an Internet of things, where trillions of devices (as well as sub-IP devices like RFID tags) are connected to the Internet. N1 Grid architectures and products position you to be prepared for the new distributed computing paradigms, where business services are created on-the-fly from loosely coupled coarse-grained shared services that live on the Internet. Service Oriented Architectures (SAOs) and other distributed computing ideas are gaining a lot of attention as a way to solve the challenges of:

- Hyper-frequent change – Driven into the data center by competitive pressures and unrelenting customer demand for new services
- Lots of users – The exploding and unpredictable nature of web access coupled with the distributed computing world where other services will now be interacting with your services
- Lots of devices – The challenges of filtering and aggregating useful information from the signals of trillions of active devices

- Lots of data – Whether harvesting from designed loads (for example, data grids between collaborating sites) or organizing oceans of data from things like customer self-service access, negotiation and filtering for localized services, and streaming video and other content to a variety of devices
- Lots of calculations – Driven by both the type and amount of data (in which interesting things reside) and the ubiquity of compute resources that can be ganged together in many different ways (for example, grids, vertical scaling, horizontal scaling, distributed object-oriented computing)

Businesses are now allowed to think of undertaking calculations never before contemplated—simulations that save research and development dollars, data mining to uncover new business information, genome and nanotech investigation and support for realtime enterprises are now not only possibilities, but regularly undertaken.

Time to Market

How many services do you release each year? What is the distribution of those release times? Exactly how have these distributions changed over the last five years? How long does it take to provision a new service into production? Is that time shorter or longer in the test environment?

Knowing the answers to these types of questions enables you to start treating IT like a business and to make more accurate projections of resources and timing. You can also break these times down even further and look for additional waste. You can set up this measurement infrastructure to enable yourself to address business issues because you know your provisioning is solid and the mobility of the applications and services you support enable you to react to issues or opportunities that arise. Think of the implications of being able to confidently tell your boss that you will be able to have six releases of a product this year versus the four released last year. Efficient design and implementation using N1 Grid architectures and products can make the IT department the trigger of bold business-changing initiatives.

Increased Availability

How many times was your environment impacted by an avoidable human error? Has that distribution changed over the last five years? How many different ways are the same servers hosting the same applications configured in your environment? How much time is spent figuring out exactly what is on a server before actual upgrade or repair work is started? Is your production environment configuration exactly equal to your test environment, and is your test environment exactly equal to your development environment? Can you guarantee that you can recreate the test environment that was in existence on a certain date? Can there be cost reduction (or better service for the same cost) in your budget due to a better distribution of skill sets if the N1 Grid architecture helps reduce complexity?

Do you expect more or less complexity in the next few years compared to the last few years? What are you doing to prepare for your answer? Fewer errors and outages let you focus on what matters in your environment—getting the environment up and keeping it up. Reducing complexity also enables you to enlist skilled resources to support new or better business initiatives because you require fewer people and less costly skill sets to handle the automated deployment of well-understood deployable entities. In addition, the simplification is achieved with solid builds and configurations can be well understood for updating or servicing. Besides the reduced outages and escalations, stability helps to reduce your help desk call volume.

Cost Measurements

Who uses what services? What is growing, and what is not growing? Have you tried to understand your costs and users? How have those costs and users changed over the last five years? Even if you do not yet use it for charge back, this utility computing type of data is useful to know and track. You can use the N1 Grid software to ease the installation of a utility computing measurement infrastructure, or to increase the utilization of your datacenter resources.

The N1 Grid vision motivates you to use good architecture and service decomposition activities. Leveraging your efforts to meter and capture use cases that influence those decomposed entities is a small change to add into the provisioning automation efforts you choose to undertake. Because the tools solution framework also maps to deployable entities in all layers of the stack, leveraging the framework for the utility computing agents and reporting structure enables you to start your efforts as small or as large as you wish. Solid builds and automated deployments reduce errors and outages, which reduce help desk call volume. The N1 Grid software products Sun is shipping today can deploy service components into a variety of operating systems (Solaris OS, Linux, IBM-AIX, and Windows), enabling your staff to leverage N1 Grid investment in architecture and process across your enterprise.

Measuring Utilization

What is your current utilization of resources? What would the business impact be if you could double, triple, or quadruple it? How many different versions of the Solaris OS, Netscape Navigator™ browser, or Oracle database do you have to deploy and support? Do you have different deployment life cycles and provisioning processes for different types of compute, storage, and network resources?

Is your business better served by you supplying a solid standard environment with enough capacity for people's services or by the application development teams spending time learning about and specifying the servers you should buy for them? Is your business better served by your team learning about and practicing with the new application's configuration files and installation mechanisms or by the developers using your standard provisioning framework and common information model naming conventions to turn over the reins to your life cycle management team when their service is smoothly deployed?

People and processes are the largest sources of cost in your data center. The N1 Grid software provides you the opportunities to install frameworks to reduce the cost and complexity. The naming conventions ensure collisions will not occur on the same compute, storage, or network resources. Sun offers several technologies that keep disparate applications and processes protected, separated, and able to count on their share of needed resources. The mobility frameworks are easy to use and simplify the movement of entities into and out of denser standardized environments. Heterogeneous environments can be provisioned using the same provisioning framework and user interfaces to the N1 Grid SPS tools.

Regulatory Measurements

How many combinations of operating systems, patch levels, and applications do you currently run? Does your testing life cycle take so long that the standard build you have been approved to use has layers dangerously out of date (no patches for six months, old firmware, not able to run to gain the benefits of new hardware)? Can you go back in time and recreate the exact build used in your modeling research environment on a certain date when you made your company's groundbreaking discovery?

New regulatory constraints often change processes and require standard and approved builds, along with other types of record keeping. Legal and intellectual property issues might necessitate the need to reproduce a research or compute environment that existed at your company on a particular day. By creating a build that the N1 Grid software deploys, you automatically provide the means to store each build. The N1 Grid architecture's separation of the stack layers means that you can maintain an application build without having to tie it to a particular operating system build. You can upgrade or work on the layers below the application (for instance, adding more compute power, storage, or network resources), while maintaining regulatory compliance by continuing to run the approved and certified application layer.

Disaster Recovery and Business Continuity

Can you redeploy your business-critical applications and keep delivering services if a disaster happens? If you fail things over to a business continuity site, can you easily redeploy in the original data center after things are fixed? How long does it take to move services to the business continuity site? Can you guarantee that what is running in the new site will be exactly what was running in the original data center? Can you use the servers in the business continuity site for your current needs until they are needed for business continuity?

N1 Grid solutions enable you to rapidly deploy services, regardless of which data center those services are deployed. You can leverage the automated mobility you use every day to provide business value to also aid in providing business continuity.

Desire to Implement N1 Grid Solutions

Not everyone is ready for the N1 Grid technology. Many people are happy to manage disks and servers and network gear. Although this type of activity is still very important and very much needed, N1 Grid solutions give you the opportunity to look at business services differently and to deliver a very different type of value to the organization. The previously discussed measurements described examples of those different CTQs, but there is another important set of measurements to make: those that outline and help others to understand and to accelerate the acceptance of the changes that N1 Grid solutions offer your business. You should use measurements to work through the items you gathered during your definition phase as you analyze your stakeholders, mobilize commitment, engage resistance, and perform other technical, political, or cultural analyses.

Change Acceptance

Quite frankly, for a change to be effective and sustainable, the need for change must exceed the resistance to it. The CTQs discussed outline the measurable benefits of a virtualized, service-centric compute environment (that is, gathering knowledge that enables IT to deliver value to a business). The famous change acceptance process (CAP) formula for the effectiveness of a change is often written as QxA=E, or quality multiplied by acceptance equals effectiveness. Hopefully, this book can convince you that N1 Grid systems have a high quality value, so that a focus on acceptance will make the change maximally effective.

Chapter 4 helped you define what the N1 Grid architecture means to you and began defining and laying out an N1 Grid solution engagement in your environment. The scenarios discussed (as well as the “CTQ Measurements” on page 9) often involved a discussion of the stakeholders and the impact and opportunities of each scenario. These types of concepts—mobilized commitment, focused leadership, compelling business need, and stakeholder analysis—are very CAP-like concepts. You know best how to match the level of CAP analysis and adherence to the magnitude of the impact you feel is warranted. Automating the application layer clearly requires much less coordination and organizational CAP than something of the magnitude of virtualizing the network and compute resources and tipping the discussion of supplied capacity to business owners from a focus on the numbers of servers and storage to that of web connections per second. You know (or your CAP analysis would discover) what is best for your environment. You know best what your competition is doing, and you know best in what order and how much you need to unleash the types of organizational opportunities and business information focus that N1 Grid solutions provide.

Benefits for People

Another way to consider other aspects of your stakeholders is to consider their early acceptance behavior. FIGURE 0-5 shows the typical acceptance behavior of stakeholders.

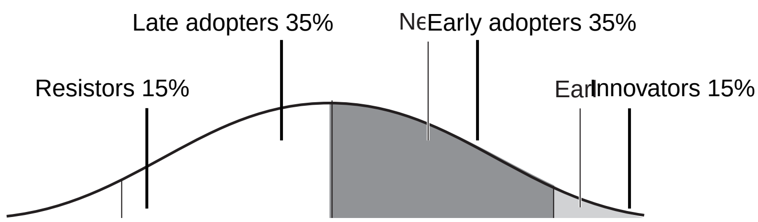

FIGURE 0-5 Typical Stakeholder Attitude Charting

Are your stakeholders early adopters, or do they prefer to wait until after a technology is released? Can your culture handle a mix of new and old provisioning, new and old focus, or new and old skill sets? Is your staff experienced in, and have they bought into, standards or repeatable frameworks? Or, is there a culture of reactive chaos and resistance to procedures, standard builds, and processes?

N1 Grid technology reduces your organization's dependency on certain types of skills, while eliminating the sources of errors from those areas. At the same time, N1 Grid solutions require a skill set that is often a natural transition for the provisioners and installers, moving those skills up a level to focus the collected knowledge and experience in the environment on business. The metrics to understand the current skill sets of your staff, the service life cycle staffing needs, and the causes and duration of downtime are some of the useful types of data to gather for analyzing the benefits to your people. Use your organization's people to help the business increase profit and to gather better data for decision making, not on provisioning applications.

Benefits for Processes

The N1 Grid vision shifts the focus of processes from those that need to protect manual activities to those that set up testable and reusable frameworks. The N1 Grid software enables the use of new processes in which the flow of information from IT enables the business to make better decisions and in which business decisions are quickly reflected in the physical environment. N1 Grid software customers often find that these new processes are created for free, through the time gained from reduced errors, automated provisioning, and efficient observability. As discussed, the metrics to understand service life cycle times, the causes and duration of outages, application provisioning times (from service request to service running, not bit-code transfer times), and business continuity considerations show the capacity to be gained. Use your organization's cycles to help the business increase profit and to gather better data, not on provisioning applications.

Benefits for Technology

The N1 Grid software uses standards and frameworks to perform activities with greater precision. Existing assets can be more densely packed because the deployable entities are well understood and can be better constrained to not impact other running entities. Mobility separates what is being deployed from where it is going to be deployed. The same assets can be used in different ways on different days, or business continuity sites can be used for day-to-day business operations until they are needed and quickly provisioned as required.

New technology, old technology, Sun technology, non-Sun technology, different versions of the same service, and separate instances of the same service can all be controlled by the same provisioning and observability systems. The type of metrics to gather and use are the waste of "silo-ed" ownership (that is, the available compute, network, and storage capacity) and the cost of owning and maintaining redundant software components that perform the same tasks for different types or families of assets.

Service Life Cycle Measurements

You can gather data to help show the quality and increase the acceptance of changes, and you can work through measurable requirements that define the desired future state. However, you must not focus only on the production environment.

Implementing a service is rarely a one-time process. The service is deployed and debugged in a development environment before it is put into a test environment. After it has been load tested and certified to be ready, it is deployed into the production environment. In addition to the changes in the lower layers (for instance, operating system changes, patches, or new, upgraded, or expanded hardware), the service often changes (for instance, patches, configuration, and tuning changes and things to compensate for or take advantage of activity in the lower layers of the stack). Metrics relating to the number and frequency of services and changes, and the time spent in the life cycle at each stage and concerning activities within each stage should be gathered. The N1 Grid software enables, and businesses require, measurements that businesses can use all through the enterprise.

The N1 Grid service life cycle model encompasses a number of high-level stages that are presented in this section using UML state diagrams (FIGURE 0-6). A state represents a specific cycle in the overall life cycle, and the transition between cycles represents a shift in the phase of the life cycle of a service. The transitions between phases in the life cycle are critically important for the following reasons:

- A service can be in only one phase of its life cycle at a time. Knowing the current phase imparts a great deal of information regarding where the service is, where it has been, and what its potential is.
- A transition is the only way for a service to migrate between life cycles.
- The requirements of a specific transition helps you understand the nature of any specific phase and its relationship with other phases.
- Transitions represent a contractual obligation with respect to the specificity of the information required to effect a transition between phases. The outputs of a phase are consumed as the input of the next phase in the life cycle.
- A preflighting mechanism can be prescribed in which the requisite information needed to affect an outgoing transition can be synthesized from templates and default values. This enables the service designer to perform what-if analysis of the service and, potentially, have various portions of the specification automatically supplied.

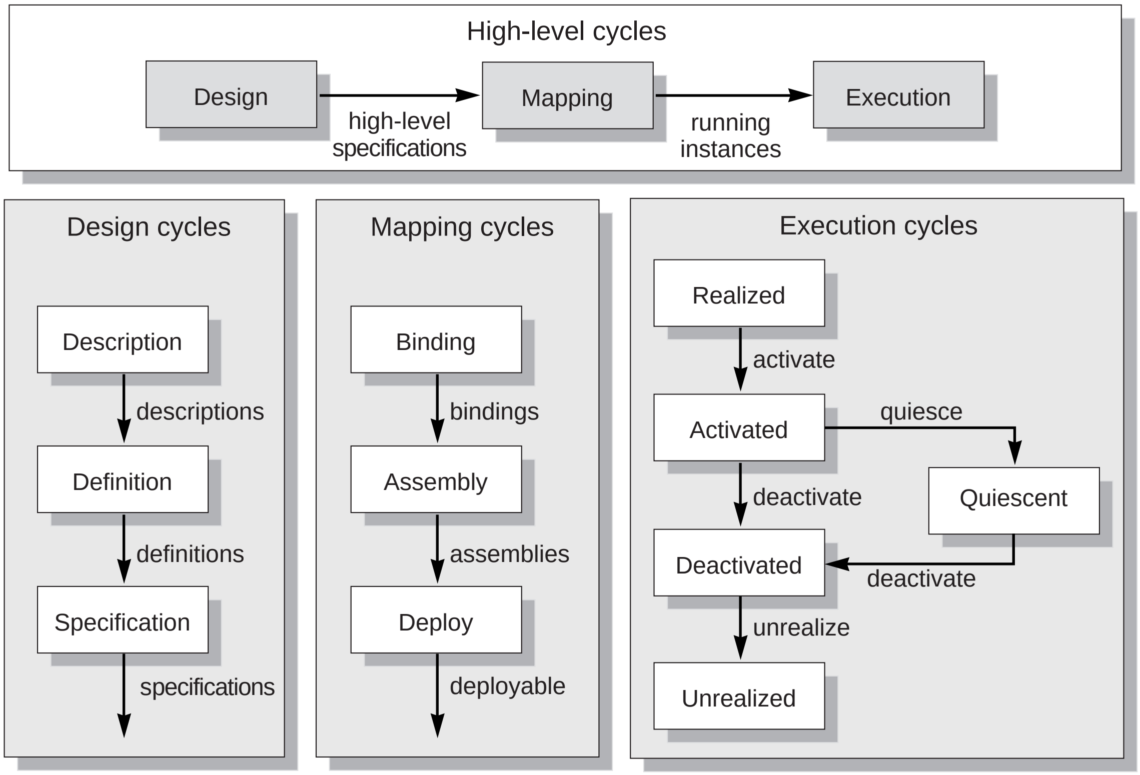

FIGURE 0-6 Example Service Lifecycle Phases

Testing Structure to Prove SLRs

Accompanying each SLR and requirement are the method to show it has been satisfied and the owner to which the SLR and requirement must be shown. Although many requirements are more binary in nature (the architecture is either an ISP with web services or it is not), many SLRs require testing harnesses and data gathering and roll-up activities (for instance, it supports 8000 web connections per second under this distribution of use cases) that must be included when considering the project effort and cost.

The data from a load test (for example, it only reached 7000 web connections per second) or the operational maturity analysis might result in changes to the original architecture. This iterative activity is a natural part of most complex architectural design methodologies.

The collection and presentation of this data takes time and disk space, but it is the only way to unambiguously demonstrate that requirements have been met. Make sure you are collecting data for all of your needs:

- Data collected to measure successful delivery on a requirement, as specified after negotiation with the owner of a requirement (for example, being asked to measure a user load maximum or a particular cluster failover time or an SNMP trap arrival time to the central console)
- Data collected to help you distinguish between possible architecture solutions and arrive at the maximal business solution (for example, articulating various scalability mechanisms to allow an economic model to also be included in the final decision)
- Data useful in a tollgate to allow transition to the next architectural methodology phase (for example, output showing successful demonstration of the use cases that prove the Jini architecture will work)

Operational Maturity Measurements

This section describes measuring the people and process aspects of your organization's operational capability. The ideas, definitions and methodology are based on Sun's Operations Management Capabilities Model (OMCM), which can describe the current state an organization's realization of the SunTone Management Framework. The tools aspects of the OMCM have been previously introduced. The authors would like to again thank Michael Moore and Edward Wustenhoff for allowing us to include an overview of their "Operations Management Capabilities Model" white paper.

The different levels of the OMCM are categorizations of an organization's service delivery capabilities. A *degree of implementation* description is used to characterize the extent to which an individual component of capability has been realized by an organization. There are five potential scores for degree of implementation:

1. Adhoc
2. Emerging
3. Functional
4. Effective
5. Optimized

Though this scoring mechanism creates a consistent terminology and simplifies the application of the model to real situations, the definition of each characteristic applies only to the component being analyzed. For example, the characteristics of a functional IT operational process are described differently than the characteristics of a functional monitoring infrastructure (for example, the characteristics that make a monitoring process optimized are very different than those that make asset management optimized).

The characteristics adhoc, emerging, functional, effective and optimized are fully defined and described for each management practice and subpractice in "Operations Management Capabilities Model." After being described, the degree of implementation for a management process can be mapped to the OMCM levels. This mapping allows the creation of a capabilities profile that describes the degree of implementation for every component at a given OMCM level. You use this profile to determine an organization's OMCM level.

People Management

A major part of delivering IT services is managing the organizations that have responsibility for executing the various IT management processes. People management describes a set of practices necessary to ensure that the IT infrastructure is staffed in an appropriate fashion and that people have the necessary skill sets. The people management practice should be a process-oriented improvement model in which the IT organization is matured through the institutionalization of different workforce management processes. The more integrated into this organization these activities become, the more effective and efficient the organization will be.

The OMCM measures capabilities that describe the degree to which people management practices have been implemented within an organization. The measuring is performed by evaluating the subpractices that comprise the five people management practices:

- Organization
- Skills development
- Resourcing
- Knowledge management
- Workforce management

The individual People Management practices and their subpractices are described below. The characteristics Adhoc, Emerging, Functional, Effective and Optimized are fully described for each practice and subpractice in the OMCM white paper.

Organization

Organizing IT services refers to activities that are related to the design of the organization's structure. These would include items such as identifying organizational groups, developing specific roles and responsibilities for each group, and describing the interfaces among groups.

The following practices are part of the organizing activity grouping:

- Communication and coordination

  This practice is focused on the establishment and maintenance of information sharing within the organization.

- Workgroup development

  This practice is focused on identifying and creating collections of individuals working together in support of specific objectives using a common, repeatable methodology.

- Workforce planning

  This practice aligns the IT organization with the business goals and objectives of the larger organization.

- Participatory culture

  This practice is focused on ensuring that decision making is performed in a structured manner and executed at the appropriate levels of the organization.

- Empowered workgroups

  This practice creates workgroups that have the responsibility and authority to determine how to most effectively conduct their operations.

- Competency integration

  This practice integrates different workforce competencies to improve the efficiency of activities that have dependencies across areas of competency.

- Organizational performance alignment

  This practice is focused on assessing how the aggregated performance of the various workgroups within the organization impact business performance.

Skills Development

Skills development is the set of activities that helps individuals acquire the knowledge and practical abilities necessary to perform current jobs or prepare them for future assignments:

- Training and development

  This practice closes the gaps between individual skills and the requirements of their current position.

- Career development

  This practice provides individuals with the opportunity to meet their career goals and objectives and is focused on continuously improving the ability of the workforce to execute the required competency based processes.

- Mentoring

  This practice facilitates the transmission of experience and knowledge throughout the organization.

Resourcing

Resourcing is the set of activities necessary to acquire the individuals to meet the goals of the organization. This would include activities to identify required skill sets, determine how many of each type is required, develop a timeline for acquiring them, and identify sources to fill the requirements. The following practices support resourcing activities:

- Staffing

  This practice matches work to individuals, including processes to recruit, select, and transition individuals into specific roles.

- Competency analysis

  This practice analyzes the business activities of the organization and develops the complete inventory of competencies needed to support them.

- Organizational capability management

  This practice manages workgroup capability to perform the competency-based processes that they are expected to use.

- Continuous capability improvement

  This practice provides the basis for supporting workgroup efforts to continuously improve the performance of their competency-based practices.

Knowledge Management

Knowledge management is the set of activities related to the capture, documentation, maintenance, and dissemination of organizational learning. Knowledge management activities enable the creation and maintenance of competency-based practices. Through the execution of knowledge management, organizations can take successful solutions and institutionalize them for reuse. This set of practices ensures that organizations move effective processes and make them repeatable:

- Competency-based practices

  This practice develops workforce competencies used to align the staffing, compensation, and other resourcing practices with the competency-development goals of the organization.

- Competency-based assets

  This practice captures the lessons learned and artifacts developed during the execution of competency-based processes, including the activities necessary to capture knowledge and disseminate this knowledge so it becomes an integral part of the organization.

- Continuous workforce innovation

  This practice drives activities necessary to set policies for workforce improvement, measure the performance of the organization against the goals, and facilitate workforce process improvement through identification of opportunities and implementation of new approaches.

Workforce Management

Workforce management is the set of activities performed to control and support individuals as they perform their tasks. This includes management of individual performance and compensation and activities necessary to provide the workforce with the infrastructure to successfully perform their job functions:

- Work environment

  This practice ensures an appropriate physical working environment so that individuals perform their job functions in an effective and efficient manner.

- Staff performance management

  This practice identifies metrics against which individual and workgroup performance can be measured. Mechanisms for rewarding superior performance are identified and formalized to reinforce the appropriate behaviors.

- Compensation

  This practice provides financial rewards to individuals in proportion to their contributions to the organization.

- Quantitative performance management

  This practice is focused on the continuous performance improvement of critical competency-based processes, which involves identifying the priority processes, developing metrics that are descriptive of the effectiveness and efficiency of these processes, and applying a process improvement methodology to performance.

Process Management

Business process management is required to support the business service life cycles—the existence and management of processes for creating, deploying, and managing IT and business services. The OMCM measures capabilities that describe the degree to which each process management practice has been implemented within an organization. The measuring is performed by evaluating the IT service subpractices that comprise the six process management practices:

- Create
- Implement
- Deliver
- Improve
- Control
- Protect

The individual process management practices and their subpractices are described below. The characteristics adhoc, emerging, functional, effective and optimized are fully described for each practice and subpractice in "Operations Management Capabilities Model."

Creating IT Services

This category describes all processes related to the creation of new services, which includes activities necessary to identify, quantify, architect, and design IT services:

- Service level management

  This process involves the planning, coordinating, drafting, agreeing, monitoring and reporting on SLAs, and the ongoing review of service achievements to ensure that the required and cost justifiable service quality is maintained and gradually improved. SLAs provide the basis for managing the relationship between the provider and the IT customer.

- Availability management

  This process manages key components of the predictability and availability of the IT services. Availability requirements heavily influence service architecture design.

Implementing IT Services

This category describes all aspects relating to the physical realization of the IT service as it is defined and created in the previous category. It addresses all aspects that ensure proper rollout of a new or updated service.

The degree of implementation is assessed by analyzing the ITIL release management process. This process protects the live environment (or IT service delivery environment) and its services through the use of formal procedures and checks. Release management works closely with the change management and configuration management processes.

Delivering IT Services

Delivering IT services is the most visible part of an IT organization's activities. This category addresses activities for the proper delivery and ongoing operation of the IT services. It is often referred to as "IT operations" or "data center operations." To assess the degree of implementation of this category, the OMCM looks at the following ITIL defined processes:

- Capacity management

  This process ensures that the capacity of the IT infrastructure matches the evolving demands of the business in the most cost effective and timely manner.

- Incident management

  This process addresses activities associated with the occurrence of service disruptions. The primary goal of the incident management process is to restore normal service operation as quickly as possible and to minimize the adverse impact on business operations.

- Service desk

  To meet both customer and business objectives, many organizations have implemented a central point of contact for handling customer, user, and related issues. The service desk is customer-facing and focused on improving service to and on behalf of the business.

Improving IT Services

This category addresses all activities surrounding the measurement and optimization of IT service activities with the goal of continuously improving service levels. To assess the level of operational capability in this category, the OMCM looks at the following processes:

- Problem management

  This process minimizes the adverse impact of incidents and problems on the business and prevents recurrence of incidents related to these errors.

  To achieve this goal, problem management seeks to get to the root cause of incidents and to then initiate actions to improve or correct the situation. The problem management process should have both reactive and proactive aspects. The reactive aspect should be concerned with solving problems in response to one or more incidents. Proactive problem management should be concerned with identifying and solving problems and known errors before incidents occur in the first place.

- Continuous process improvement

  Although ITIL understands the need for continuous process improvement, it has not defined a separate discipline to address this important aspect. The SunTone Management Framework (STMF) uses the processes as defined by Sun today; however, any Six Sigma based approach will most likely have sufficient rigor and commitment to sufficiently address this area.

Controlling IT Services

This category addresses activities to deliver the IT service within the constraints identified by the governing body, including the processes that facilitate the IT governing activities. Examples of governing functions are financial controls, audit, and alignment with business objectives.

To assess the level of operational capability in this category, the OMCM looks at the following processes:

- IT financial management

  This process reflects activities that control the monetary aspects of the business. It supports the organization in planning and executing its business objectives and requires consistent application throughout the organization to achieve maximum efficiency and minimum conflict.

- Configuration management

  The configuration management process provides a logical model of the infrastructure or a service by identifying, controlling, maintaining, and verifying the versions of configuration items in existence.

- Change management

  This process standardizes methods and procedures for efficient and prompt handling of all changes to minimize the impact of change-related incidents on service quality. The basic concepts of change management are principally process related and managerial, rather than technical (whereas incident management is primarily technical, with a strong emphasis on the mechanical nature of some of the processes).

Protecting IT Services

This category addresses all activities that ensure that IT services are still available under extraordinary conditions such as catastrophic failures, security breaches, or unexpected heavy loads. As businesses depend more and more on IT services, this area becomes more and more important to address.

To assess the degree of implementation of this category, the OMCM looks at the following ITIL defined processes:

- IT service continuity management

  This process supports the overall business continuity management process by ensuring that the required IT technical and services facilities (including computer systems, networks, applications, telecommunications, technical support, and service desk) can be recovered within required, and agreed, business time constraints.

- Security management

  ITIL defines security management as the process of managing a defined level of security on information and IT services. Included is the reaction to security incidents. Security management is more than physical security and password disciplines. It includes data integrity, confidentiality, and availability. Security management is not an isolated process. It is part of IT and business. The relationship between security management and the other ITIL processes is such that each process has the obligation to perform the required security tasks wherever possible. These tasks in each ITIL process should address the security aspects in their specific area, but the point of control of these tasks is centralized by the security management process. Security management is governed by a corporate policy that drives budget, focus, and management direction. Within ITIL practices, this information is normally found in the service level agreements.

Policy Measurements

This chapter has discussed metrics and useful data for provisioning and observability. Policy combines those functions into a feedback loop that anticipates, corrects, and improves your business. There are many opportunities to observe and react to events in your data center. Examples of measurements to consider include:

- Deployable service and container relationships

  As discussed in Chapter 4, a policy can use information describing the capabilities of containers and the requirements of a deployable service to perform matches, check for health, and provide information to other entities interested in this type of N1 Grid architecture relationship.

- Infrastructure

  Changes to business services affect the infrastructure upon which they run. Instrumenting infrastructure components and providing policies to assist with the scalability, security, and tuning of infrastructure components is a necessary step in moving to the mobile, adaptable data center that the N1 Grid enables.

- Business continuity

  Now that the N1 Grid software makes configuration and reconfiguration quick and effortless, measuring for the necessary triggers to kick off business continuity can inform the N1 Grid software policy and accompany a revisiting of the business continuity initiatives. Disaster recovery sites can be filled with resources doing useful business until the trigger comes to inform the N1 Grid software that it is time to configure the site for business continuity.

Planning for a policy can start immediately, solidifying the measurements to guide the N1 Grid software policy. It is worthwhile to organize a policy model, begin populating the information model, and start to connect the business and system viewpoints. The N1 Grid vision, architecture, and products can help inform this effort.

Security Measurements

Just because the N1 Grid software flawlessly automates the installation of your approved and hardened golden services does not automatically mean that someone cannot log onto one of your servers and make undesired changes. N1 Grid solutions require the same vigilance, defense in depth, and change control that your environment deserves today, but it offers several additional capabilities that make some of that work easier.

Many N1 Grid software products support security processes by making it possible to compare the current environment against the approved and hardened golden image that is expected to be running in a particular location. Clearly, it is beneficial to be able to easily identify unexpected deltas to your builds. The roles used by the N1 Grid software can also segment access and entitlements. They divide not only along lines of operation (for instance, only certain roles can create services and only certain roles can install them), but also along access and control within business groups or within services in a given group. You can best choose how to parse out the identity and entitlements needed to control and run your business, but the N1 Grid software can help you leverage the value and safety out of a strong identity infrastructure.

Mobility is another area where policy, security, and a common information model combine to provide safety and efficiency. Having identity and rules to guide what entities can (and cannot) reside in common locations or in particular combinations speeds up the process to deployment or to react to a business need or observed and reported situation. A secure environment runs with data. Information must be collected and presented to enable command and control of who, when, and under what conditions an action is taken in the N1 Grid operating environment. Your security officer should have clear requirements for the confidentiality, integrity, and availability of the data in your data center and a clear picture of the security architecture and security operations that combine to minimize risk. Sun's layered security architecture approach elevates security to a systemic quality that must be viewed holistically. Data comes from many individual layers. FIGURE 0-7 shows an example of a possible layered security architecture.

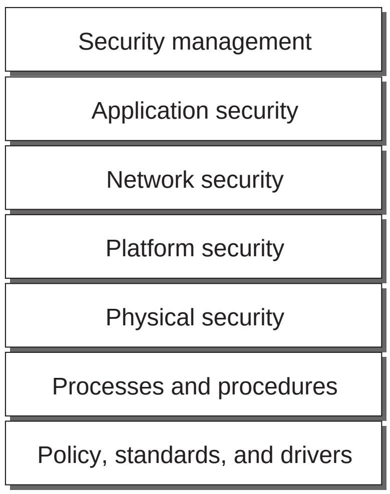

FIGURE 0-7 Layered Security Architecture Example

Each layer represents a required area of discussion, action, and instrumentation, but equally important are the connections between layers and the capability of the stack as a whole. Security is a systemic quality and must be viewed holistically, and its CTQs must be identified, measured, and rolled up appropriately for reporting.

Summary

The N1 Grid architecture accelerates capability and frees up IT organizations to focus on business needs. Policy and security impart the checks and balances to ensure that the actions and requests are properly controlled and channeled, but they require accurate information and enterprise process control to function.

This chapter described the types of measurements needed to translate customer needs and requirements into measurable characteristics (CTQs) and to understand the capabilities needed to collect, roll up, analyze, and use the CTQs and the N1 Grid business information throughout the service life cycle. Methods to measure and analyze CTQs were seen to support both the situations where advancement to the next phase of the architectural methodology is being considered, as well as to prepare for where ongoing observation of production environment must take place.

The examples should have helped you strengthen the requirements and SLRs captured for your N1 Grid solutions. The N1 Grid architecture enables new types of CTQs to be measured and delivered, enables input from an entirely new set of business stakeholders, and enables IT to align even closer to what the business needs. Use cases and SLRs can be elevated to support service life cycles and business policy over traditional compute, network, and storage activities.

Traceability provides the key to gathering, analyzing, validating, and verifying the requirements that you will use in architecture analysis and solution construction. A set of final requirements is only a beginning step to a final architecture and running solution because the architecture methodology should be open to accept new information from the architecture analysis and solution verification phases.

You should now be equipped to establish the measurable success criteria for your N1 Grid software engagement. Chapter 6 discusses the next steps to take: analyzing the prioritized requirements, constraints, and assumptions of your business in preparation for an N1 Grid architecture and design.

About the Authors

Jason Carolan is a Senior Architect and Principal Engineer at Sun Microsystems. He has spent most of his five years at Sun in Sun Client Services, focused on developing solutions for Sun's customers, systems architecture, and improving architectural quality through patterns. Jason joined the N1 Grid software team over two years ago, contributing to many of the early internal and external documents. He has also been responsible for the design of the Service Delivery Network Architecture, which is Sun Professional Services^{SM} program's data network architecture standard—one of the best examples of modular architecture. He also speaks regularly at conferences around the World about network design, security, and the N1 Grid architecture. Jason lives in Colorado where he enjoys the amazing snow and sun.

Scott Radeztsky is one of Sun's Principal Engineers. His six years at Sun have been focussed on architecture, performance and tuning, education, and working with customers to navigate uncharted architecture, technology, and operations territory in novel solution deployments. As examples of this, Scott has had the unique opportunity of directly architecting and building some of Sun's first internal and external customer-based N1 Grid software environments and is now focused in the areas of edge computing and Sun's next generation of systems. Before joining Sun, Scott earned a Ph.D. in Particle Physics from the University of Wisconsin, using the Web and related technologies for analysis and collaboration since their inception. Scott is grateful for the patient support and understanding of his family during this book creation effort.

Paul Strong is a Systems Architect at Sun Microsystems, working in the N1 Grid software product group. He has spent most of his seven years at Sun focused on resource management and availability and joined the N1 Grid software team at its inception, over three years ago. This is Paul's first official authoring credit, although he is an uncredited contributor to the Sun BluePrints book on resource management. He also speaks regularly at conferences. He is a native of England, transplanted to California, is a keen musician, and holds a degree in Physics, specializing in Astrophysics, from the Victoria University of Manchester, England.

Ed Turner is an Architect in Sun's Technology and Solutions Organization, focused on supporting major customers in the telecommunications industry. Leveraging his seven plus years of working various technical and management roles at Sun, he focuses his efforts on guiding his customer's strategic architecture. Ed had the unique opportunity of directly architecting and building some of Sun's first significant customer-based N1 Grid software environments. When not spending his off hours with his wonderful family, Ed enjoys long distance motorcycle touring. Ed resides in the Atlanta, GA, area and has a BSCS degree from the University of Maryland.

Ordering Sun Documents

The SunDocs[SM] program provides more than 250 manuals from Sun Microsystems, Inc. If you live in the United States, Canada, Europe, or Japan, you can purchase documentation sets or individual manuals through this program.

Accessing Sun Documentation Online

The `docs.sun.com` web site enables you to access Sun technical documentation online. You can browse the `docs.sun.com` archive or search for a specific book title or subject. The URL is `http://docs.sun.com/`

To reference Sun BluePrints OnLine articles, visit the Sun BluePrints OnLine web site at: `http://www.sun.com/blueprints/online.html`