Non-Functional Requirements (NFR) Framework

A Subset of the Enterprise Architecture Framework

A White Paper by: Rajesh Radhakrishnan

Sr. Managing Consultant/Sr. IT Architect
IBM Global Technology Services

September 2009

This White Paper is an informational document and does not form part of the TOGAF documentation set. Readers should note that this document has not been approved through the formal Open Group Standards Process and does not represent the formal consensus of The Open Group Architecture Forum.

Non-Functional Requirements (NFR) Framework

Document No.: W098

Published by The Open Group, September 2009

Any comments relating to the material contained in this document may be submitted to:

The Open Group
44 Montgomery St. #960
San Francisco, CA 94104

or by email to:

ogspecs@opengroup.org

Comments on the content of this document may be submitted to the author at:

rajesh.radhakrishnan@us.ibm.com

Table of Contents

Boundaryless Information Flow™
achieved through global interoperability
in a secure, reliable, and timely manner

Executive Summary

This White Paper focuses on Non-Functional Requirements (NFR) for IT and IT-enabled business services and proposes the creation of enterprise architecture artifacts specifically addressing NFR. It describes an NFR lifecycle and framework.

One of the major goals of enterprise architecture planning, development, and implementation is the alignment between business goals and objectives and IT capabilities. As business strategy and architecture (Phase B of the TOGAF Architecture Development Method (ADM)) drives IS and IT architecture and capabilities (Phases C and D of the ADM), the center piece of all architecture efforts and projects is requirements management.

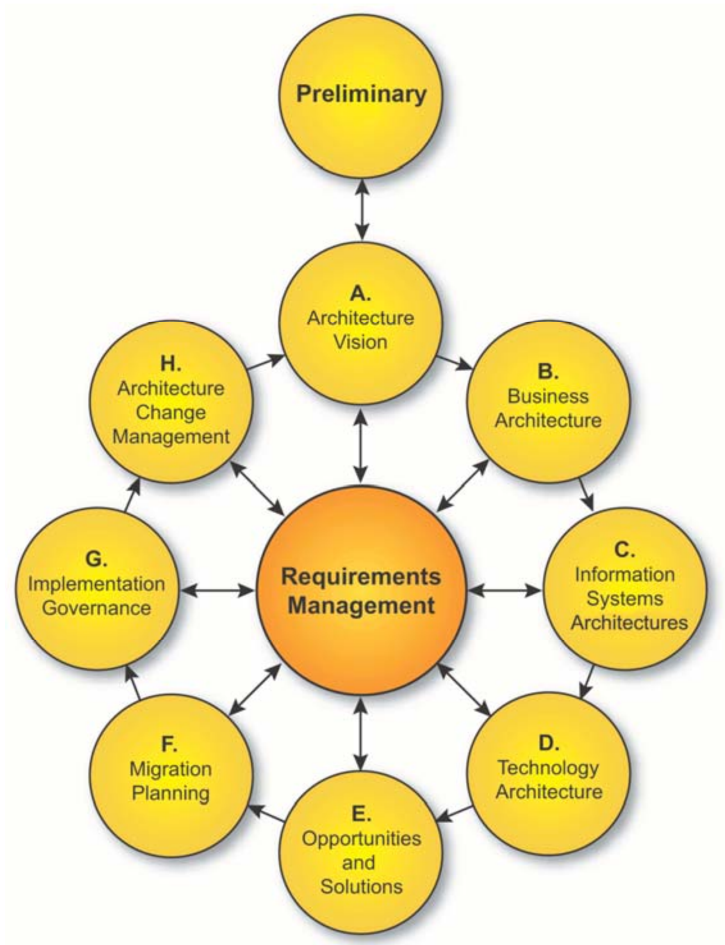
As enterprise architecture groups typically develop a set of artifacts such as building blocks, patterns, and reference architectures, among others, artifacts associated with requirements become key and re-usable assets for several enterprise architecture projects.

Service functional requirements vary greatly from industry to industry and from one business function to another. While there is variation in service non-functional requirements, there are opportunities to create enterprise-specific standard templates for non-functional requirements such as:

- A set of NFR dimensions prioritized or rank-ordered
- A set of resiliency tiers, with each tier having specific targets for resiliency metrics
- A standard set of data security requirements for customer or employee data
- A standard set of requirements associated with compliance to an industry regulation, etc.

Introduction

Functional and non-functional requirements are the center piece of most enterprise architecture frameworks including TOGAF™ Version 9.[1] All phases of the TOGAF 9 Architecture Development Method (ADM) relate to requirements and requirements management.



Quotes from TOGAF 9:

- "Every stage of a TOGAF project (architecture projects that comply with the TOGAF ADM) is based on and validates business requirements."
- "Requirements are identified, stored, and fed into and out of the relevant ADM phases, which address, prioritize, validate, and dispose of these requirements".

This White Paper organizes requirements into functional and non-functional requirements and focuses on non-functional requirements (NFR). The NFR Framework impacts (has potential benefits for):

- Enterprise architecture projects

[1] The Open Group Architecture Framework (TOGAF), Version 9 is available at www.opengroup.org/architecture/togaf. It is referred to in the rest of this White Paper as "TOGAF 9".

- Enterprise technology domain architecture & design projects (such as enterprise storage architecture projects)
- Business service(s) architecture & design projects (for example, CRM service)
- IT service(s) architecture & design projects (for example, email service)

The ADM in TOGAF 9 has been applied to security.[2] The ADM for security is actually the ADM applied for security architecture and security requirements. Security itself is a non-functional requirement. Similarly, the ADM can be applied to other non-functional requirements dimensions such as availability, continuity, and efficiency, among others.

An organization publishing its NFR Framework document as an enterprise architecture artifact and as part of its enterprise architecture will help the enterprise with:

- Identifying a set of non-functional requirements dimensions as service qualities and sub-qualities that are relevant to the enterprise
- Prioritizing these non-functional requirements dimensions from a business perspective; i.e., generic enterprise-level prioritization which helps with specific service architectures – individual service-level non-functional requirements prioritization may use this document as a guideline and can vary the service-specific prioritization scheme
- Documenting methods to arrive at non-functional requirements, which helps the enterprise to re-use and apply methods to arrive at service-specific non-functional requirements
- Re-using non-functional requirements-related metrics and metric target levels for each IT and IT-enabled business service
- Map business objectives to non-functional requirements, non-functional requirements to specifications (technology domain and process specifications), and to non-functional requirements-related metrics/metric targets
- Developing service tiers (or service classes) based on non-functional requirements-related metrics and target levels; service tiers, or service classes would be another enterprise architecture artifact where most enterprise IT and IT-enabled business services are bucketed into one of these service tiers via a service rationalization process
- Mapping enterprise architecture and enterprise domain architecture patterns, building blocks, and products to service tiers (with varying non-functional requirements)

[2] Refer to TOGAF 9, Chapter 21, Security Architecture and the ADM.

Stakeholders for Non-Functional Requirements

Senior management and the corporate board generally provide strategic guidelines, directives, and objectives which can be translated to high-level non-functional requirements. As a case in point, after events such as September 11 in New York and Hurricane Katrina in New Orleans, several enterprises have upped their priority for business continuity and have had directives from senior executives and the board to develop business and IT service continuity capabilities and, in fact, have set business and IT recovery objectives which easily translate to service (business & IT service) continuity requirements.

The business unit and business service owners also act as customers of IT and IT-enabled business services and do state high-level objectives which can translate to requirements.

The business owners, sometimes, also delegate authority to customer liaisons (as buyers of IT services) and they represent end users to provide input for IT and IT-enabled business service requirements. The customer liaisons set the requirements for an IT or IT-enabled business service and represent the business owners, the end users, and other service stakeholders. The liaisons provide input for service requirements from a business, end-user, and regulatory perspective, among others.

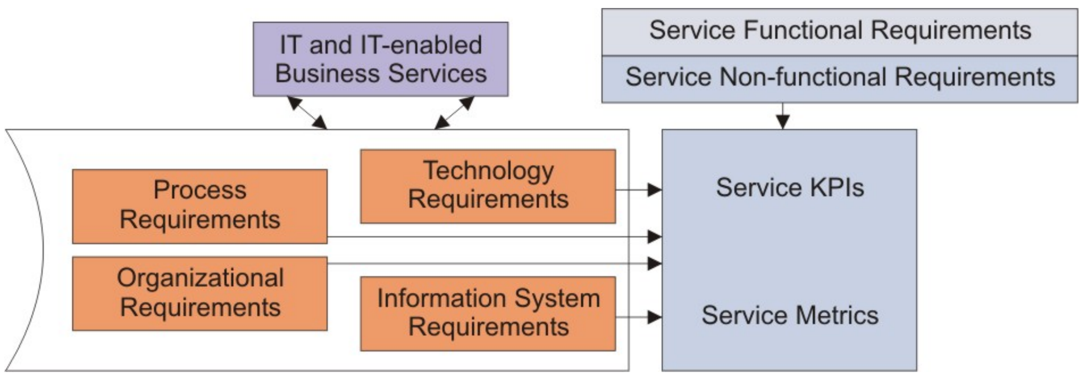
The customer is part of the business and sets expectations (requirements) while the requirements management team (architects) and service-level management team from IT help gather, define, refine, and document a set of funded requirements and define solutions that can meet these documented requirements.

Non-Functional Requirements for What?

This White Paper focuses on non-functional requirements for IT and IT-enabled business services. Service non-functional requirements are essentially the same as service qualities. Services (both IT and IT-enabled business services) are designed, developed, transitioned, managed, and improved using:

- Information system (application, integration, and data) architecture and capabilities
- Technology (network, server, storage, among others) architecture and capabilities
- Process (business & IT processes) architecture and capabilities
- Organization (business & IT) architecture and capabilities

Service requirements also need to be translated to information systems (IS), information technology (IT), process, and organizational requirements. This translation process, which helps with requirements mapping and traceability, is further discussed in NFR Lifecycle.



Most business and IT environments have built IS, IT, process and organizational capabilities. Therefore, in most cases, business & IT have to work together to leverage current capabilities and tune them to meet or exceed service functional and non-functional requirements. For example, a requirement to restore a run-time service (the service quality being Service Recoverability) within 30 minutes after a service outage implies such technology requirements as development, replication, and maintenance application images and server images for fast recovery, application, and server clustering for fast failover, standby db for fast failover, and so on. The same 30-minute time to restore service requirements implies advanced event and incident monitoring and management requirements (process requirements) such as less than three minutes to detect an incident, less than three minutes to escalate, and so on. Information system requirements can be that all application components should allow for watchdog capabilities for automated application component fault detection, restart, and recovery. Organizational requirements can be a 24 by 7 support team with less than three-minute response time as part of an Operational Level Agreement (OLA) within the enterprise and as part of support contracts with the service-related vendors.

Service Non-Functional Requirements and Service Lifecycle

Service non-functional requirements (or service quality requirements) impact the entire service lifecycle from service design, to development, transition, operation, and improvement (ITIL v3 Service Lifecycle stages).

Service Design and Development

Traditionally, service design and development focused on functional requirements while service operations focused on non-functional requirements. However, with the advent of service quality-related Integrated Development Environment (IDE) tools and technologies, service qualities – such as service availability, performance, and security – are being factored into the service design and development lifecycles. The IT industry has made major progress in the last few years in terms of pulling forward non-functional requirements into the service lifecycle.

As organizations fund service design and development efforts that focus on non-functional requirements, the up-front investment made on the service development cycle is likely to be paid off via improved service resilience and performance against non-functional requirements. For example, developing service components that comply with the Distributed Management Task Force (DMTF)/Service Availability Forum (SAF) Hardware Platform Interface (HPI) and Application Interface Specification (AIS) can surely improve service availability and minimize the costs associated with service unavailability. Similarly, J2EE applications that comply with JSR 88 Live Application Upgrade capabilities will require less downtime for maintenance windows, thereby minimizing costs associated with planned or scheduled application maintenance windows.

Applications designed to work with enterprise event monitoring and management capabilities can prevent a set of service incidents by leveraging event correlation for predicting service and service component failures.

Advanced service security capabilities designed into the service – such as automated service-related change auditing and reporting capabilities – can reduce security-related service incidents in production.

Advanced service-specific state monitoring and capturing capabilities – such as the DTrace utility for Solaris (UNIX®) services – can help with quick root cause analysis, problem management, and control of problem-related costs (for example, labor hours spent on Root Cause Analysis (RCA)).

Service Transition

Service transition focuses on service verification, testing, and validation while preparing services for production readiness. There are several verification, testing, and validation capabilities focused on non-functional requirements. However, due to time-to-market (services) issues and aggressive schedules, service transition activities are frequently rushed through. This results in bringing services to production without appropriately testing and validating non-functional requirements-related service capabilities.

Currently, most organizations focus service testing and validation cycles over certain non-functional requirements, such as load testing as part of service capacity and performance management. Resiliency testing (such as availability and continuity testing) is left for the service operations phase of the service lifecycle.

Most organizations and industry bodies recognize the need to beef up service transition capabilities as most organization recognize that their service transition capabilities are the weakest part of their service lifecycle capabilities.

However, prioritization of service qualities or non-functional requirements dimensions at the enterprise level (by business and enterprise architecture teams) and at the service level can help service teams to develop specific testing and validation capabilities for those non-functional requirements dimensions that are of the highest priority.

Also, the higher the level of non-functional requirement – i.e., the higher the service-level requirements – the higher the level of service transition work including verification, testing, and validation. For example, the higher the level of required service availability, the more the need for testing of specific service availability building blocks, such as clustering software or autonomic systems.

In several cases, the non-functional requirements-specific building block – such as clustering and availability monitoring software for an application – may very well be a set of vendor products well-tested and proven by the vendor. However, the primary testing by the organization buying these non-functional requirements building blocks would be integration testing (integration with the overall solution) and testing to validate the claims of the vendor when it comes to the organization-specific service implementation (of the vendor building block). However, this level of integration testing and testing to validate capabilities against non-functional requirements and vendor claims are key for identifying potential scenarios via test cases where the overall solution may not meet the non-functional requirements.

Service Operations

Infrastructure building blocks which host service building blocks are a key part of meeting non-functional requirements in the production or operation phase of a service lifecycle.

A sample set of infrastructure building blocks linked to a set of services is shown below:

- Clustered servers hosting web services
- Grid network hosting VoIP services
- Content Addressed Storage (CAS) hosting content management application
- Storage Area Network (SAN) hosting multimedia content services
- Clustered appliance hosting DNS services

Just as infrastructure building blocks and their specifications are important for meeting or exceeding non-functional requirements in the operational environment, operational processes – such as event, incident and problem management, production configuration, change, release, and deployment management – are also key for non-functional requirements.

Service design processes such as availability, continuity, performance/capacity, and security management processes also play a significant role in the production environment when it comes to service availability, continuity, performance/capacity, and security.

Service Improvement

Continuous Service Improvement (CSI) and Service Improvement Plans (SIP) involve services, such as:

- Availability improvement plans
- Performance/capacity improvement plans
- Utilization improvement plans

- Continuity improvement plans
- Security improvement plans
- Cost improvement plans
- Usability improvement plans

Each plan aligns with a service quality (non-functional requirements dimension) and leverages patterns, building blocks, emerging and new technologies that improve performance with regard to one or more service quality (non-functional requirements dimension).

Arriving at Requirements

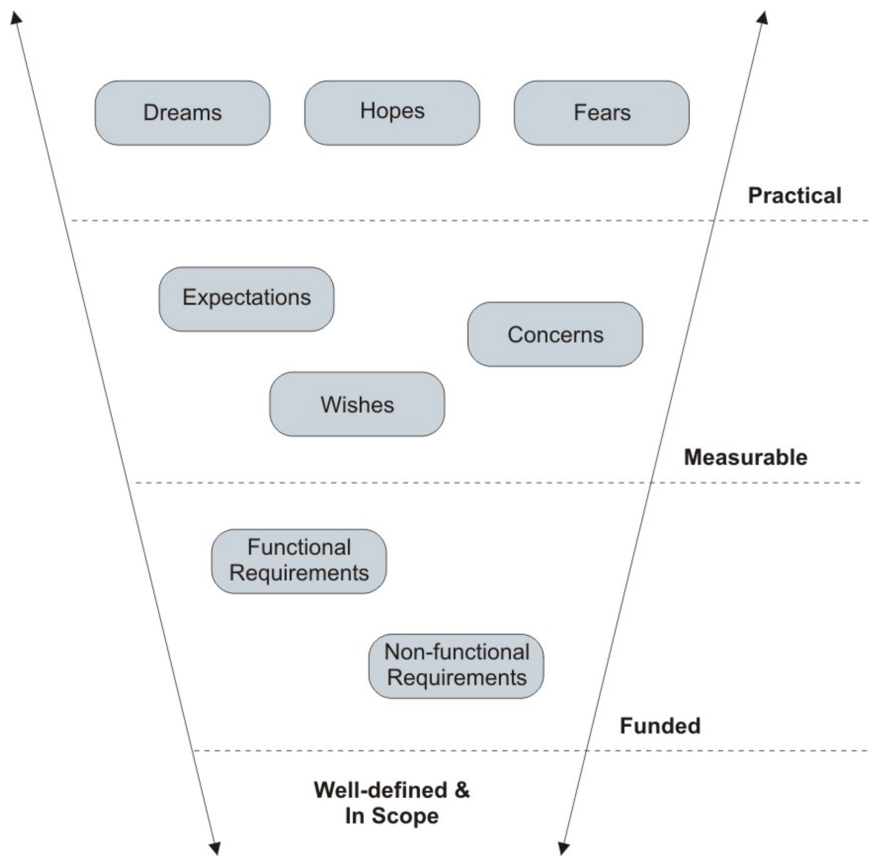
Both functional and non-functional requirements can be elicited from different stakeholders via:

- Workshops
- Focus groups
- Interviews
- Story-boarding (scenarios)

and can logically flow (be distilled) from:

- General stakeholder hopes, dreams, and fears to
- Practical expectations, wishes, and concerns to
- Measurable requirements to
- Funded in-scope and documented functional and non-functional requirements

This method to arrive at requirements is graphically shown below.



Service Quality Dimensions (Service Non-Functional Requirements)

While the dimensions of functional requirements vary significantly from one service to another, the dimensions of non-functional requirements need not vary as much. Therefore, an enterprise service NFR Framework with a baseline of service non-functional requirements, associated capabilities (patterns, building blocks), and associated metrics can be developed at the firm and industry level and made part of the enterprise architecture, enterprise technology domain architecture, standards, governance, and service management. These industry and organization-specific non-functional requirements artifacts can drive requirements management for most if not all IT and enterprise architecture projects. This White Paper covers a generic (industry-agnostic) NFR Framework.

Dimensions of service non-functional requirements or service quality are:

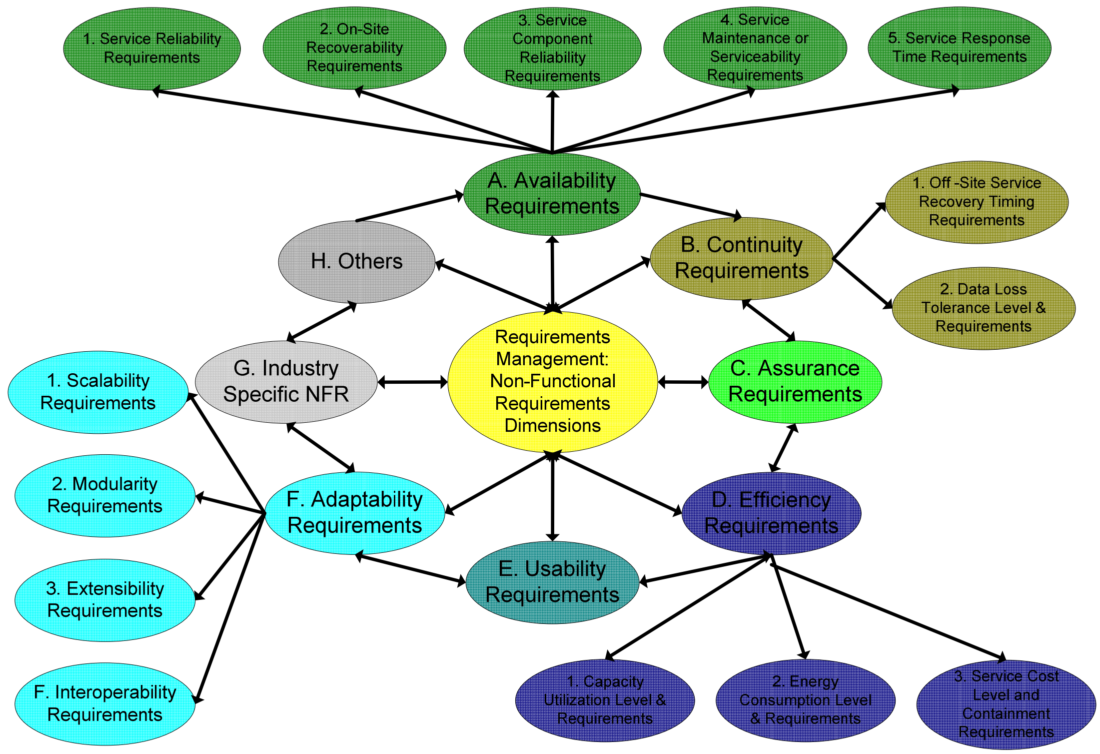
- **Service Availability (& Continuity)**: reliability, manageability, serviceability, performance (response time), recoverability, and continuity, among others
- **Service Assurance**: security, confidentiality, integrity, credibility, non-repudiation, and data protection, among others
- **Service Efficiency**: cost of service per unit, utilization, and service activity monitoring, among others
- **Service Usability**: ease-of-use, locatability, accessibility, and locale (international operations capability), among others
- **Service Adaptability**: interoperability, scalability, portability, modularity, and extensibility, among others

Each service quality (non-functional dimension) and its sub-qualities has a set of metrics associated with it and the target level for this metric becomes the service requirement or service-level requirement.

For example, Service Reliability is a Service Availability sub-quality and two key metrics associated with Service Reliability are Mean Time Between Service Failure (MTBSF) and Mean Time Between Component Failure (MTBCF) (service component). The MTBCF for a hard drive as a service component can range from months to years. The target of five years (or 60 months) of operations before disk failure, for a set of disk drives that are part of a service (say email), is a requirement (design specification) associated with a service component.

Similarly, serviceability is a Service Availability sub-quality and can have several metrics associated with the same. The total hours-of-service maintenance window and the number of service maintenance window per year are key metrics. A service can have a requirement of 24-hour maintenance window once per quarter which implies a maximum of 96 hours of maintenance window per year. So the serviceability requirements are a maximum of four maintenance windows per year which require scheduled service outage and a maximum of 24 hours per service maintenance window.

Service component (resource) utilization is a service efficiency metric and the target of 80% logical disk space utilization level is a requirement. Virtualization and dynamic logical disk scaling (scaling up or down) technologies, combined with resource monitoring technologies, allow for managing resource utilization levels in real time.



(Adapted from TOGAF Service Qualities)

Important Note: Several of these key service qualities and sub-qualities have one or more IT processes that map to them in such IT process frameworks as ITIL. Examples are service design processes such as service availability management, service continuity management, service performance & capacity management, and service security management, among others.

Prioritizing Non-Functional Requirements

Several firms treat availability, continuity, and security as the most critical non-functional requirements and classify them as such. The remaining non-functional requirements dimensions are important but treated as secondary when compared to availability, continuity, and security. The prioritization scheme may vary from one industry to another and one organization to another.

As a case in point, an online brokerage firm competing on the basis of cost and value to its customer can treat service efficiency – especially from a cost structure perspective – as critical to the firm's strategy and competitive advantage. As such, efficiency becomes a key non-functional requirements dimension both for domain architectures and service architectures. In fact, the online brokerage firm's IT department has a dedicated and collaborative process to constantly look for opportunities for cost-cutting (cost take-out) and implements them faster than its competition. This enterprise was among the first in the industry to replace several of its high-end UNIX systems with low-end Linux systems in the 1990s. The penetration of low-cost Linux systems in the firm's data centers is also highest for the industry.

An enterprise's business and IT strategy, business model, and industry and regulatory environments are some of the factors that determine the prioritization scheme.

NFR Lifecycle

Each service quality or service non-functional requirement has a certain lifecycle which starts with requirements definition and gathering to realizing the defined and funded requirements.

Analysis and Planning for Non-Functional Requirements

Analysis and planning for non-functional requirements produces service-level objectives and service-level requirements. Also, it produces (IS and IT) domain-level objectives and requirements. Analysis and planning for non-functional requirements includes leveraging an organization-specific NFR Framework and artifacts (which helps with requirements definition) and initial gathering of non-functional requirements. Certain service-specific non-functional requirements can be arrived at via planning and analysis activities. As a case in point, service continuity requirements can be arrived at via business impact analysis, service impact analysis, and risk analysis – all three analytical activities are part of service continuity management in the ITIL IT process framework. Similarly, cost of service unavailability analysis can help the organization arrive at service availability requirements. These analytical methods that produce service requirements documentation are typically part of an IT process and involve stakeholders discussed earlier (see Stakeholders for Non-Functional Requirements).

The table below shows the mapping of specific analytical methods that lead to specific service quality objectives and non-functional requirements.

Planning & Analysis Leads to Service Quality Objectives and Non-Functional Requirements

→

Cost of Uniavailability, Service Failure Impact Analysis (SFIA), Component Failure Impact Analysis (CFIA), Fault Tree Analysis (FTA), Failure Mode Effect & Capability Analysis (FMECA)	Availability
Risk Analysis, Business Impact Analysis (BIA), CCTA Risk Analysis & Management Method (CRAMM)	Continuity
CCTA Risk Analysis & Management Method (CRAMM), Consultative, Objective, & Bi-functional Risk Analysis (COBRA)	Assurance
Service Cost Structure Analysis, Business Process Re-engineering (BPR) Analysis, Energy Consumption Analysis	Efficiency
Ease-of-Use Analysis, Service Acceptance Analysis, Service Accessiblity Analysis	Usability
Business Model & Operating Models, Business Modularity Analysis, Service Modularity Analysis, Compatibility Analysis	Adaptibility

Business scenarios are key outputs of analytical work associated with non-functional requirements. These business scenarios can have associated use-case scenarios. As an example, there are several business scenarios associated with business continuity events, security events, among others. The same is true with scenarios with regard to business service availability. The scenario analysis gets more complicated with outsourcing of certain business and IT processes and services. However, artifacts documenting business scenarios and use-case scenarios associated with non-functional requirements can benefit the enterprise from a planning perspective, particularly when these artifacts are re-used and applied when different groups are engaged in documenting non-functional requirements related to the service they are developing or maintaining.

Architecture for Non-Functional Requirements

Architecture for non-functional requirements refines non-functional requirements and develops architecture specifications. Service architecture specifications (email architecture specifications) and domain architecture specifications (network architecture specifications), are some of the key outputs of this phase. Non-functional requirements-related use-case scenarios and use-cases are also key outputs of this phase. Mapping of non-functional requirements to standards are also part of this phase. Domains include application, data, and integration (IS) domains as well as network, server, storage, facilities, systems management, desktop, and other such technology domains. Architecture for non-functional requirements covers the process of evaluation and selection of a set of design patterns and building blocks that have the potential to implement a specific sets of non-functional requirements. Architecting for non-functional requirements also includes the process of refining non-functional requirements and mapping of non-functional requirements to specific architecture building blocks.

Engineering for Non-Functional Requirements

Engineering, design, and build specifications (non-functional requirements-related specifications) are key outputs of this phase. These specifications can be service-specific (such as email specifications) or at the domain level (such as middleware specifications). Test case scenarios and test cases are also key outputs of this phase which helps in traceability of non-functional requirements from gathering to realization. Non-functional requirements engineering, design, and build as well as testing help with the implementation of solutions that meet or exceed non-functional requirements. Non-functional requirements engineering includes:

- Non-functional requirements engineering, such as performance engineering
- Design for non-functional requirements, such as designing for availability
- Building with non-functional requirements building blocks
- Testing for each non-functional requirements test case

Standards-related specifications are a key part of this phase and the next phase. Several standards organizations provide documented specifications that are very relevant for re-use during the engineering and operating phases of the NFR Lifecycle.

Operating for Non-Functional Requirements

Operating an IT or IT-enabled business service at the agreed service levels implies certain operational capabilities that include:

- IS and IT configuration parameters (technology capabilities)
- IT process specifications and capabilities
- IT organizational specification and capabilities

An organization realizes its service quality objectives and non-functional requirements when the service is in operation and there are service operational capabilities that help to meet or exceed the service non-functional requirements on a day-to-day basis.

Non-Functional Requirements-Related Monitoring

Each non-functional requirement or service quality dimension has a set of sub-qualities and metrics associated with each sub-quality. How does the service in question perform against the service quality targets (non-functional requirements) set for the service? This question can be answered with the Enterprise Monitoring and Management (EMM) architecture and capabilities. EMM tools can help with business, service, and resource monitoring and reporting. Measurable and reportable metrics that map to the service qualities or non-functional requirements are key for non-functional requirements monitoring. SMART metrics are Specific, Measurable, Achievable, Relevant, and Time-bound.

Note: Non-functional requirements themselves can be performance targets for certain key metrics. For example, Mean Time to Restore Service (MTTRS) is a key metric for Service Availability (quality) and Service Recoverability (sub-quality), and number and percent of time the 30 minutes MTTRS is met or exceeded for email as a service becomes a measurable and reportable non-functional requirements metric for email.

Metrics models (managed services and managed resources metrics models) such as the Distributed Management Task Force (DMTF) CIM Metrics Model with its Unit of Work (UoW) definition, Base Metric definition, and Base Metric value provide a standards-based model for defining and managing non-functional requirements-related metrics. As per the CIM Metrics Model, there can be several units of work such as batch jobs, user-initiated interactive operations, completed and committed transactions, and so on associated with a service. There are also several metrics associated with each UoW. UoW metrics and measurements are at a more granular level than service metrics. A UoW can have several associated non-functional requirements metrics, even though the CIM Metrics Model-related UoW metrics are primarily time taken to complete the UoW (performance dimension) and status of the UoW (availability dimension).

Non-Functional Requirements-Related Improvements

In ITIL parlance, Continuous Service Improvement (CSI) and Service Improvement Plans (SIP) involve measures to improve service qualities and service capability to meet or exceed non-functional requirements. For example, service outage analysis and the application availability patterns result in availability improvement plans, particularly when IT organizations face Service-Level Agreement (SLA) breaches due to unplanned outages. Similarly, IT organizations can have improvement plans for each service quality or sub-quality discussed in this White Paper.

Also, new and emerging technologies, processes, and organizational capabilities can directly improve certain service qualities. Examples include replication technologies and the exponential decline in the cost of storage space (disk space measured in gigabytes), which has allowed for significant improvements in service continuity capabilities. Another example is grid storage that offers improved storage performance & resilience capabilities over older storage methods.

The NFR Lifecycle results in the realization and improvement of a service performance when it comes to non-functional requirements, as shown in the table below.

NFR Lifecycle	Architecture Management	Example: Service Continuity (as an NFR)
Phase 1: Planning & Analysis for NFRs	Executive Directives & Board Objectives	Business Continuity in an hour after Event/Disaster
Phase 2: Architecting for NFRs	Funded/Documented NFRs	Service RTO & RPO of one hour or less
Phase 3: Engineering for NFRs	Engineering & Design Specifications	Active-Passive Service Cluster across Data Center for Applications & Servers, asynchronous replication every 15 minutes for service data
Phase 4: Operating for NFRs	Configuration Parameters & Process Specifications	Service DR exercises every three months, Service DR imrovement plans after every exercise
Phase 5: NFR-related Monitoring	SMART Metrics & Data	% of services meeting RTO, % of services meeting RPO (during DR exercises)
Phase 6: NFR-related Improvement	Static & Dynamic Thresholds	Services exceeding RTO by 10%, Services exceeding RPO by 10% (e.g., if RTO is 60 minutes, service takes more than 66 minutes to recover)

The six phases of the NFR Lifecycle discussed above can apply to each of the service quality dimensions discussed in this document. Developing and maintaining artifacts associated with each of these phases (for each service quality dimension) can bring about efficiencies in individual service lifecycle, via application, modification, and re-use of these artifacts.

The NFR Lifecycle indicates a logical flow (deductive reasoning) from generic (broad objectives) to specific (configuration parameters and specifications) which help with requirements mapping and tracing.

Summary

In general, the activities and inputs/outputs of each phase map to the different architecture disciplines:

- Phase 1 maps to the business architecture as well as the business & IT process architecture, as several of the analytical methods used in Phase 1 are part of a business and IT process and their outputs can be used as artifacts for the business architecture.
- Phase 2 maps to the business architecture and service architectures that are embedded in the enterprise (and enterprise architecture). There are enterprise-level requirements documents and service-specific requirements documents.
- Phase 3 maps to the IT and IS architecture primarily in terms of re-use of non-functional requirements building blocks which implement design specifications.
- Phase 4 maps to the IT operations architecture from a configuration management and IT process management perspective.
- Phase 5 maps to the Enterprise (or Event) Monitoring and Management (EMM) architecture which includes resource, service and business monitoring, and reporting capabilities (associated with functional and non-functional requirements).

- Phase 6 maps to the IT operations architecture and specifically data-driven (non-functional requirements-related performance data) Service Improvement Plans (SIP) and capabilities.

The lifecycle as it is applied to service performance and capacity is discussed below.

Service Performance and Capacity

Phase 1: Performance and Capacity Analysis and Planning

Phase 1 includes:

- Historical analysis of capacity, utilization, and performance data at the business, service, and resource level
- Gathering and documentation of performance-related objectives and requirements
- Development of capacity and performance strategy
- Development of capacity and performance tiers (if applicable) – as part of service-level tiers
- Developing and documenting a set of relevant and applicable performance architecture and design patterns – across technology domains (e.g., caching which can be used at the storage, network, application, and other domains) and within technology domains (e.g., fast-reader lanes for the application/presentation logic domain)

Phase 2: Architecture for Performance

Phase 2 includes leveraging performance-related architecture patterns and building blocks – such as load balancing, aggregation (virtualization-based or otherwise), clustering, and caching, among others. More details about performance-related patterns by domain are available in the paper "Anatomy of Architecture" by Mahesh Mohta (see References).

Phase 3: Engineering for Performance

Phase 3 includes leveraging performance design patterns and practices by domain:

- Software performance engineering
- Middleware performance engineering
- Database performance engineering
- Network performance engineering
- Operating system performance engineering
- Server/network/storage hardware performance engineering
- SAN performance engineering

The areas covered are modeling for performance, developing for performance, performance (load) testing, and validation, among others.

Phase 4: Performance and Capacity Operations

The operations-related work for performance and capacity management include:

- Operational readiness review from a capacity, utilization, and performance perspective

- Review and validation of configuration parameters with regard to performance-related specifications by domain (for example, are the DB and table spaces partitioned as per the specifications? Are the row isolation levels appropriate enough to prevent too many DB lock-related performance issues? And so on).

Phase 5: Performance Monitoring

Standards-based (such as Application Response Measurement (ARM) from The Open Group) and non-standards-based application response time and other response time monitoring tools (such as seek time for storage devices), bandwidth (capacity), throughput (utilization), and latency (performance) measurement tools, by technology domain that capture, record (say for a technical data warehouse or PCM database), and report are a critical part of managing for performance. While Phases 1 through 4 help with the definition phases of Six Sigma[3] DMAIC (Define, Measure, Analyze, Improve, and Control), Phase 5 deals with measurements and monitoring of data associated with measurements.

Performance monitoring also includes:

- Performance measurement and metrics (that map to requirements)
- Streaming and real-time data associated with each performance measurement
- Data based on real and synthetic end users (pseudo-users)
- Thresholds and dynamic thresholds, among others

Phase 6: Performance Improvement

Systematic study and analysis of performance-related data in production, including:

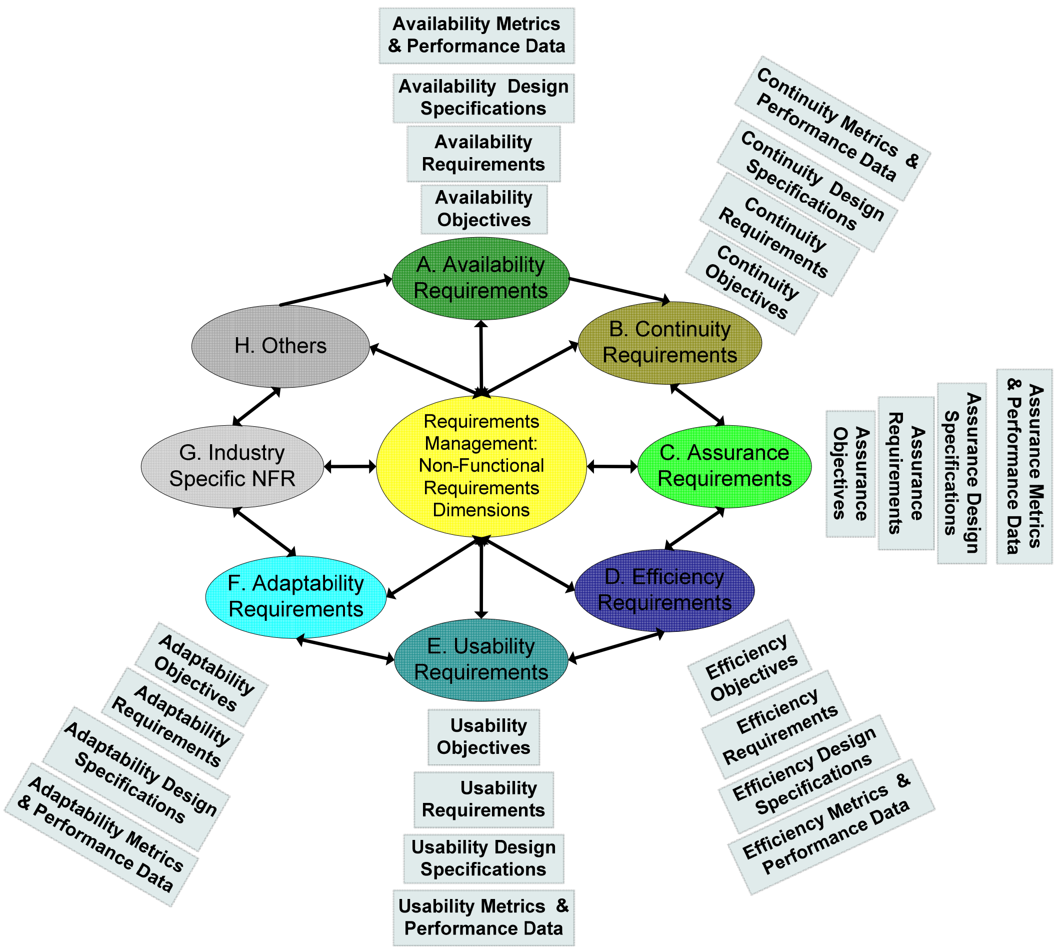
- Correlation analysis
- Gap analysis with best practices
- Gap analysis with target performance architecture
- Lack of relevant design patterns and building blocks
- Process analysis

among others, should result in performance improvement plans to better control performance of applications and tools from an end-user and business perspective.

[3] Six Sigma is a business management strategy, initially implemented by Motorola, that today enjoys widespread application in many sectors of industry.

NFR Framework

At the core, the NFR Framework has to cover the identification and prioritization of non-functional requirements dimensions and management of the NFR Lifecycle for each dimension. The NFR Lifecycle and the key outputs of the NFR Lifecycle are objectives, requirements (documented and funded non-functional requirements), specification, metrics, and non-functional requirements-related performance data.



Service tiers can further improve the NFR Framework by defining a set of service non-functional requirements per service tier. The application or service rationalization process should help classify applications into a predefined service tier. Each service tier will have a set of non-functional requirements-related capabilities (technology, process, and organizational capabilities). Defined service tiers and well-defined service tier-related capabilities are also part of the enterprise architecture repository which can be re-used for service lifecycle activities.

As the NFR Framework is part of an enterprise architecture framework, there is a one-to-one mapping of NFR Lifecycle to service-level management, as a process domain in IT service management.

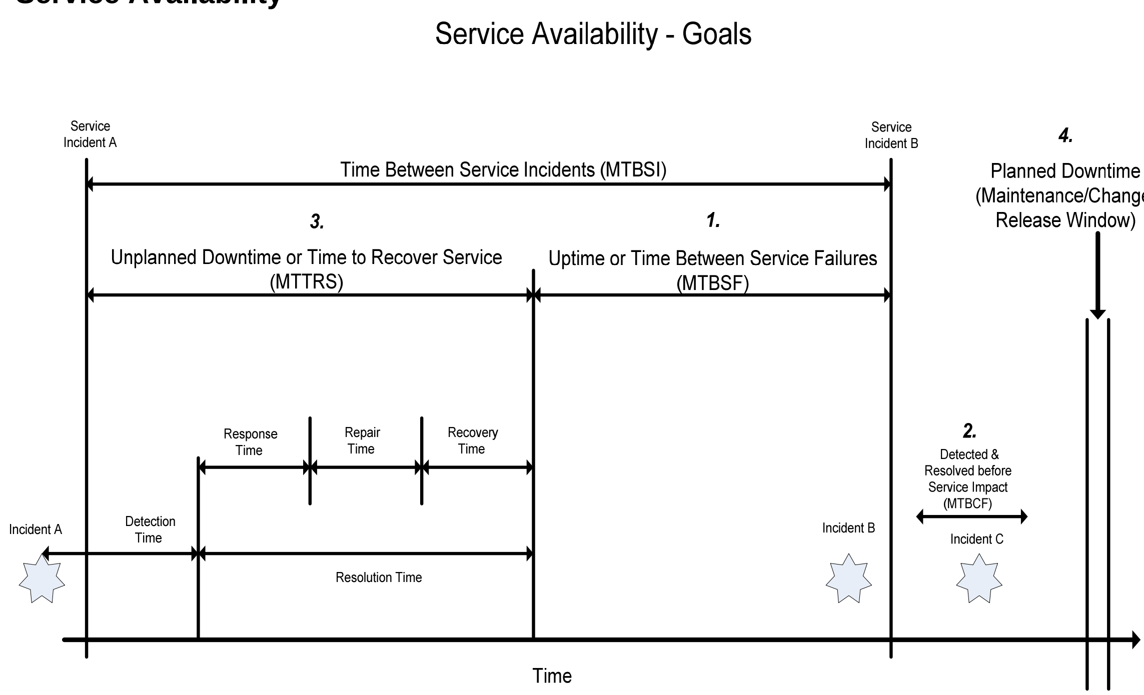
As the table below indicates, column 1 refers to the NFR Lifecycle and column 2 refers to some of the key outputs of the NFR Lifecycle at the enterprise level. Column 3 maps the same outputs at the service level. As much as possible, re-use of artifacts that make up column 2 and are outputs of column 1 (NFR Lifecycle and enterprise architecture process work), for the service-level management activities for each IT and IT-enabled business service, brings about efficiencies in the organization. Service-level management outputs at column 3 can significantly benefit from the application/customization and re-use of the Architecture Management outputs in column 2.

NFR Lifecycle	Architecture Management	Service-Level Management
Phase 1: Planning & Analysis for NFRs	Executive Directives & Board Objectives	Service-Level Objectives
Phase 2: Architecting for NFRs	Funded/Documented NFRs	Service-Level Requirements
Phase 3: Engineering for NFRs	Engineering & Design Specifications	Service Design Specifications
Phase 4: Operating for NFRs	Configuration Parameters & Process Specifications	Service Configuration Parameters
Phase 5: NFR-related Monitoring	SMART Metrics & Data	Service KPIs & Service-Level Monitoring
Phase 6: NFR-related Improvement	Static & Dynamic Thresholds	Thresholds for Service KPIs

While the Architecture Management outputs (column 2) drive (to a significant degree) enterprise architecture, the Service-Level Management outputs (column 3) drive (to a significant degree) service architectures. Enterprise architecture would cover the enterprise as a whole and directly influence enterprise domain architectures (enterprise horizontals such as enterprise network architecture or enterprise storage architecture). Service architecture would cover services as a whole (such as enterprise email service or enterprise CRM service) and leverage and use enterprise domain architecture (and capabilities).

Details about a couple of non-functional requirements dimensions – Service Availability and Service Continuity and associated sub-qualities – are discussed in the following sections.

Service Availability

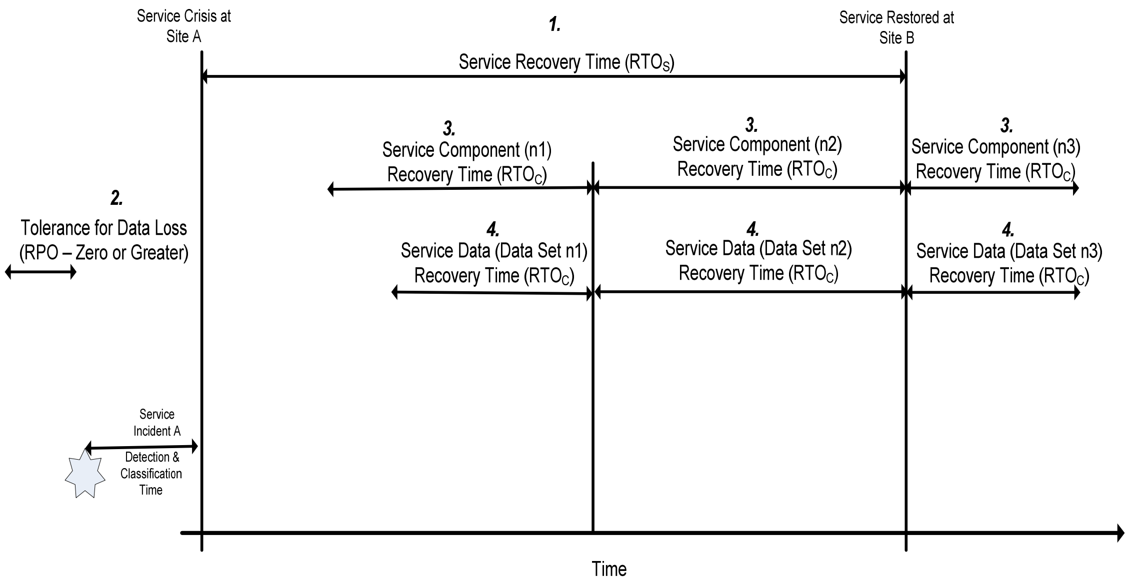


Service Availability is a function of Service Reliability as measured by Mean Time Between Service Failure (MTBSF), Mean Time Between Component Failure (MTBCF), and the ability to contain component failures with minimal or no impact to Service Availability, Service Recoverability (operational recovery *versus* disaster recovery) as measured by Mean Time to Restore Service (MTTRS), and the ability to maintain Service Availability while performing service maintenance and changes; i.e., serviceability.

Service Recoverability as a sub-quality of the Service Availability dimension and a set of sample specification, parameters, and metrics are shown below.

NFR Lifecycle	**Architecture Management**	**Service Availability (as an NFR)**
Phase 1: Planning & Analysis for NFRs	Executive Directives & Board Objectives	Set target of four 9 availability for key services
Phase 2: Architecting for NFRs	Funded/Documented NFRs	Service Recovery is less than 15 minutes (Operational Recovery *vis á vis* Disaster Recovery)
Phase 3: Engineering for NFRs	Engineering & Design Specifications	Active-Passive Service Cluster within Data Center for Applications, DB, and Servers, Fault Detection & Watchdog for every service component with Watchdog Specifications
Phase 4: Operating for NFRs	Configuration Parameters & Process Specifications	Cluster compliance with SAF specifications
Phase 5: NFR-related Monitoring	SMART Metrics & Data	Service Uptime, Service Component Uptime, MTTRS, MTTR Service Component, # of Service Failures per year, # of Service Component Failures per year, etc.
Phase 6: NFR-related Improvement	Static & Dynamic Thresholds	Service & Service Component Failover Time exceeding three minutes, # of Service Components that failed more than once in a year

Service Continuity



Service Continuity is dependent on the capabilities associated with the time-to-recover service (after an event/incident/crisis), tolerance level for data loss (due to an event/incident/crisis), often referred to as recovery point, and time-to-recover service components (such as servers, network, application components, db, and so on) and time-to-restore or recover data at the continuity site.

Service recovery time and recovery point objectives and a set of sample specifications, parameters, and metrics are shown below.

NFR Lifecycle	**Architecture Management**	**Service Continuity (as an NFR)**
Phase 1: Planning & Analysis for NFRs	Executive Directives & Board Objectives	Business continuity in an hour after event/disaster
Phase 2: Architecting for NFRs	Funded/Documented NFRs	Service RTO & RPO of 1 hour or less
Phase 3: Engineering for NFRs	Engineering & Design Specifications	Active-Passive Service Cluster across Data Center for Applications & Servers; Asynchronous replication every 15 minutes for service data
Phase 4: Operating for NFRs	Configuration Parameters & Process Specifications	Service DR exercises every 3 months; Service DR improvement plans for every exercise
Phase 5: NFR-related Monitoring	SMART Metrics & Data	Percentage of services meeting RTO; Percentage of services meeting RPO (during DR exercises)
Phase 6: NFR-related Improvement	Static & Dynamic Thresholds	Services exceeding RTO by 10%; Services exceeding RPO by 10% (e.g., if RTO is 60 minutes, service takes more than 66 minutes to recover)

References

- The Open Group Architecture Framework (TOGAF), Version 9; refer to www.opengroup.org/architecture/togaf9-doc/arch
- Distributed Management Task Force (DMTF), Common Information Model (CIM) Metrics Model; refer to www.dmtf.org/standards/documents/CIM/DSP0141.pdf
- Distributed Management Task Force (DMTF)/Service Availability Forum (SAF), Hardware Platform Interface (HPI) and Application Interface Specification (AIS); refer to www.dmtf.org
- Information Technology Infrastructure Library (ITIL); refer to www.itil-officialsite.com
- Application Response Measurement (ARM), The Open Group; refer to www.opengroup.org/management/arm
- IT Service Management for High Availability, IBM Systems Journal; refer to www.research.ibm.com/journal/sj/474/radhakrishnan.pdf
- Enterprise Architecture and IT Service Management, White Paper published by The Open Group; refer to www.opengroup.org/bookstore/catalog/w078.htm
- Using ITIL Best Practices to Create a Capacity Management Process, Chris Molly, IBM (for CMG); refer to www.cmg.org/measureit/issues/mit08/m_8_3.html
- Anatomy of Architecture, Mahesh Mohta, Infosys (for CMG); refer to www.cmg.org/measureit/issues/mit54/m_54_1.html and www.cmg.org/measureit/issues/mit55/m_55_1.html

About the Author

Rajesh Radhakrishnan is a Senior Managing Consultant and Senior IT Architect for IBM Global Technology Services (GTS). His current client engagements involve global ITIL program management, service-level management, and service availability/continuity management. Prior to joining IBM, he was an independent IT architect at Zygous, LLC for two years and a senior technical consultant at Sun Microsystems, Professional Services, for nearly five years. He was also a senior technical staff member at Metamor Worldwide for five years and consulted for numerous Fortune 500 companies. He specializes in enterprise architecture and service management consulting, business continuance, high-end systems, high availability, storage technologies, and technical project management.

Rajesh has an MBA from the Old Dominion University and an MS degree from the University of Virginia, McIntire School of Commerce. He was awarded the VITS (Virginia IT Scholar) for 2003. He is certified as a Solaris Systems Administrator, Storage Architect, TOGAF practitioner, ITIL-ITSM consultant (manager's level), IBM Senior IT Architect, and as an Open Group Master Architect.

Rajesh Radhakrishnan
IBM Global Technology Services, 12174 Chancery Station Circle, Reston, VA 20190, USA
rajesh.radhakrishnan@us.ibm.com
rajmit.blogspot.com

About The Open Group

The Open Group is a vendor-neutral and technology-neutral consortium, whose vision of Boundaryless Information Flow™ will enable access to integrated information within and between enterprises based on open standards and global interoperability. The Open Group works with customers, suppliers, consortia, and other standards bodies. Its role is to capture, understand, and address current and emerging requirements, establish policies, and share best practices; to facilitate interoperability, develop consensus, and evolve and integrate specifications and Open Source technologies; to offer a comprehensive set of services to enhance the operational efficiency of consortia; and to operate the industry's premier certification service, including UNIX® system certification. Further information on The Open Group can be found at www.opengroup.org.