IBM GLOBAL SERVICES

G08

Availability Management Concepts for an On Demand World

Mike Bonett, IBM Corporation

bonett@us.ibm.com

zSeries Expo | Nov. 1 - 5, 2004

Miami, FL

RETURN TO INDEX

Trademarks

Those trademarks followed by ® are registered trademarks of IBM in the United States, other countries, or both; all others are trademarks or common law marks of IBM in the United States, other countries, or both.

- AF/OPERATOR®
- AF/REMOTE®
- AIX®
- AS/400®
- BladeCenter
- Candle Management Server®
- CandleNet®
- CandleNet Portal®
- CICS®
- CICSPlex®
- Database 2
- DB2®
- Domino®
- e-business(logo)®
- e(logo)e-business®
- e-business (logo)
- e-business on demand
- eServer
- Enterprise Storage Server®
- GDPS®
- Geographically Dispersed Parallel Sysplex
- HACMP
- HACMP/6000
- IBM®
- ibm.com®
- IBM TotalStorage Proven
- IMS
- MQSeries®
- MVS
- MVS/ESA
- MVS/XA
- Netfinity®
- Netfinity Manager
- NetView®
- OMEGACENTER®
- OMEGAMON®
- OMEGAMON II®
- OMEGAVIEW®
- OMEGAVIEW II®
- OS/390®
- OS/400®
- Parallel Sysplex®
- RACF®
- Redbooks
- S/390®
- System/390®
- ThinkPad®
- Tivoli®
- Tivoli (logo)®
- Tivoli Enterprise
- Tivoli Enterprise Console®
- Tivoli Management Environment®
- TotalStorage®
- VM/ESA®
- VTAM®
- WebSphere®
- z/OS®
- z/VM®
- zSeries®

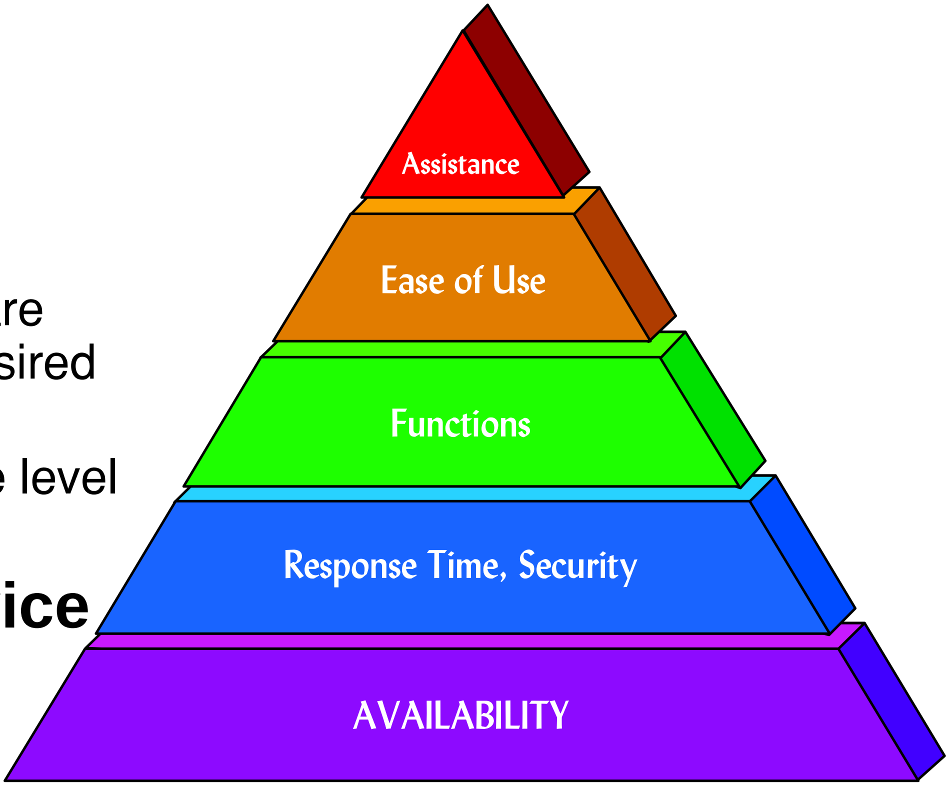
Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.
Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.
Intel, Intel Inside (logos), MMX and Pentium are trademarks of Intel Corporation in the United States, other countries, or both.
UNIX is a registered trademark of The Open Group in the United States and other countries.
Linux is a trademark of Linus Torvalds in the United States, other countries, or both.
Other company, product, or service names may be trademarks or service marks of others.

Agenda

- ★ **What is "AVAILABILITY" and why is it important?**
- ★ **What does it mean within the On Demand context?**
- ★ **What design techniques are used to provide and improve availability?**
- ★ **How should it be measured - including from an On Demand perspective?**
- ★ **How does one plan for implementing, monitoring, and improving availability?**

Traditional Views of Availability

- **Physical**
  - The state in which a component can be used for its intended purpose
- **User**
  - The state in which productive work can be accomplished
- **Application (end-to-end)**
  - Combines the physical and user view
    - All components that support the application are allowing applications users to perform the desired functions
  - Normally the basis for customer/provider service level agreements
- **The foundation for application service**
  - Access to business logic
  - Access to business information
  - The primary way to assess the quality of service provided

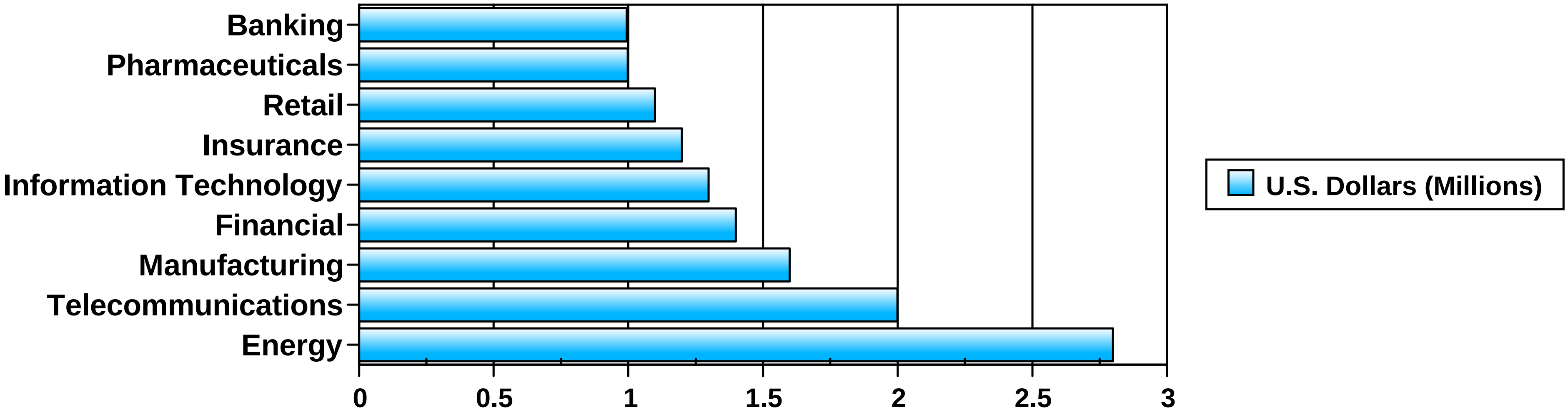


Unavailability is a Real Expense

- **Business Revenue**
  - Based on transaction business value
  - Can also be lost "revenue opportunities"
- **Productivity**
  - Application/Business System users (idle and recovery time)
  - Support personnel/systems (resolving unavailability situations)
  - Based on the personnel cost of those impacted
- **Brand**
  - Negative publicity
  - Lost customers
  - Supplier relationships
- **Regulatory**
  - Fines
  - Penalties
- **Legal**
  - Contractual Obligations (and associated penalties)
  - Late fees
  - Litigation

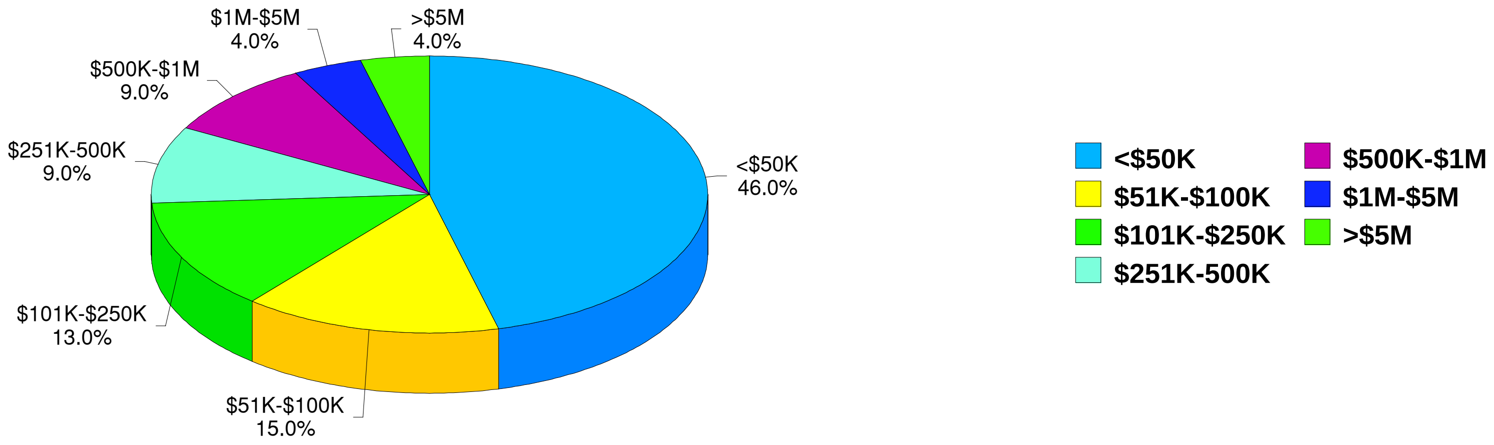
The Cost of Unavailability

Average Hourly Impact on Businesses



Source: Meta Group, October 2000

Hourly Cost of an Outage - Survey



Source: Eagle Rock Alliance, LTD., 2001

What Is Your Cost?

- Information required to calculate costs:
  - Cost per employee
  - Application/business system revenue
  - Revenue per transaction
  - Transactions per time period
  - Outage information
    - Frequency
    - Length
    - Impacted employees and workload transactions
  - Percentage of employee/revenue impact
- Remember intangible costs
- Essential for understanding the investment level required to address exposures

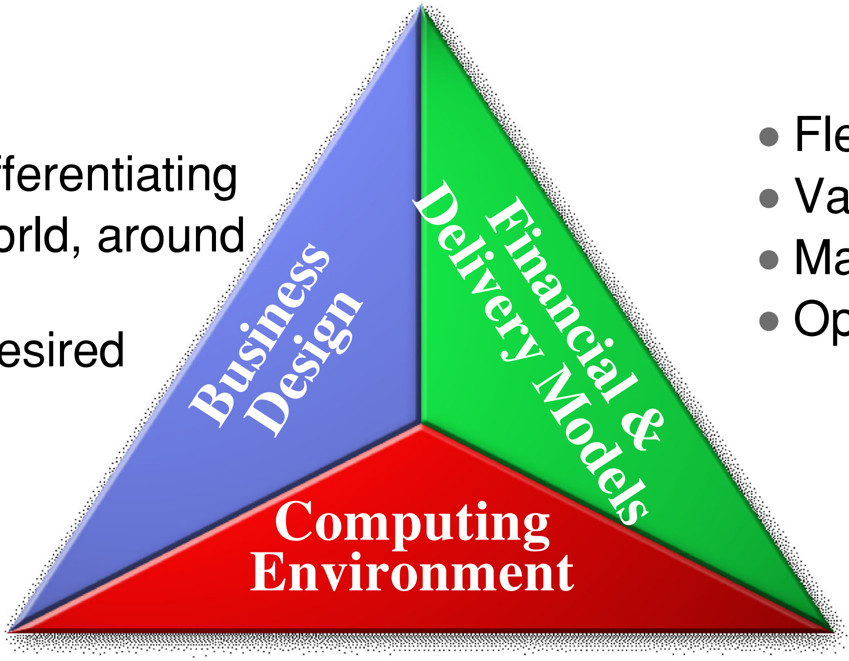
Cost Example

Application annual revenues	$50,000,000
Expected annual hours of operation	5,616 (18 x 6 x 52)
Revenue per hour	$8,903
Employees impacted	300
Employee cost per hour	$50
Employee impact per hour	$15,000
Hourly Application Outage impact	$23,903
Cost at 99% availability (56.16 hours)	$1,342,392
Cost at 99.5% availability (28.08 hours)	$671,196
Hourly impact - at 50% revenue, 30% employee impact	$8,953
Cost at 99% availability (56.16 hours)	$502,744
Cost at 99.5% availability (28.08 hours)	$251,372

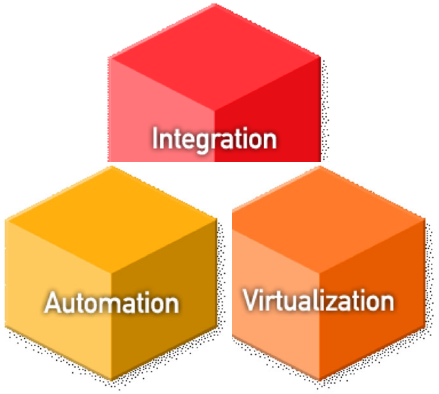
The On Demand Environment

An enterprise whose business processes – integrated end-to-end across the company and with key partners, suppliers and customers – can respond with flexibility and speed to any customer demand, market opportunity or threat

- Responsive in real time
- Variable cost structures
- Focused on core and differentiating
- Resilient - around the world, around the clock
  - AVAILABILITY is a desired characteristic

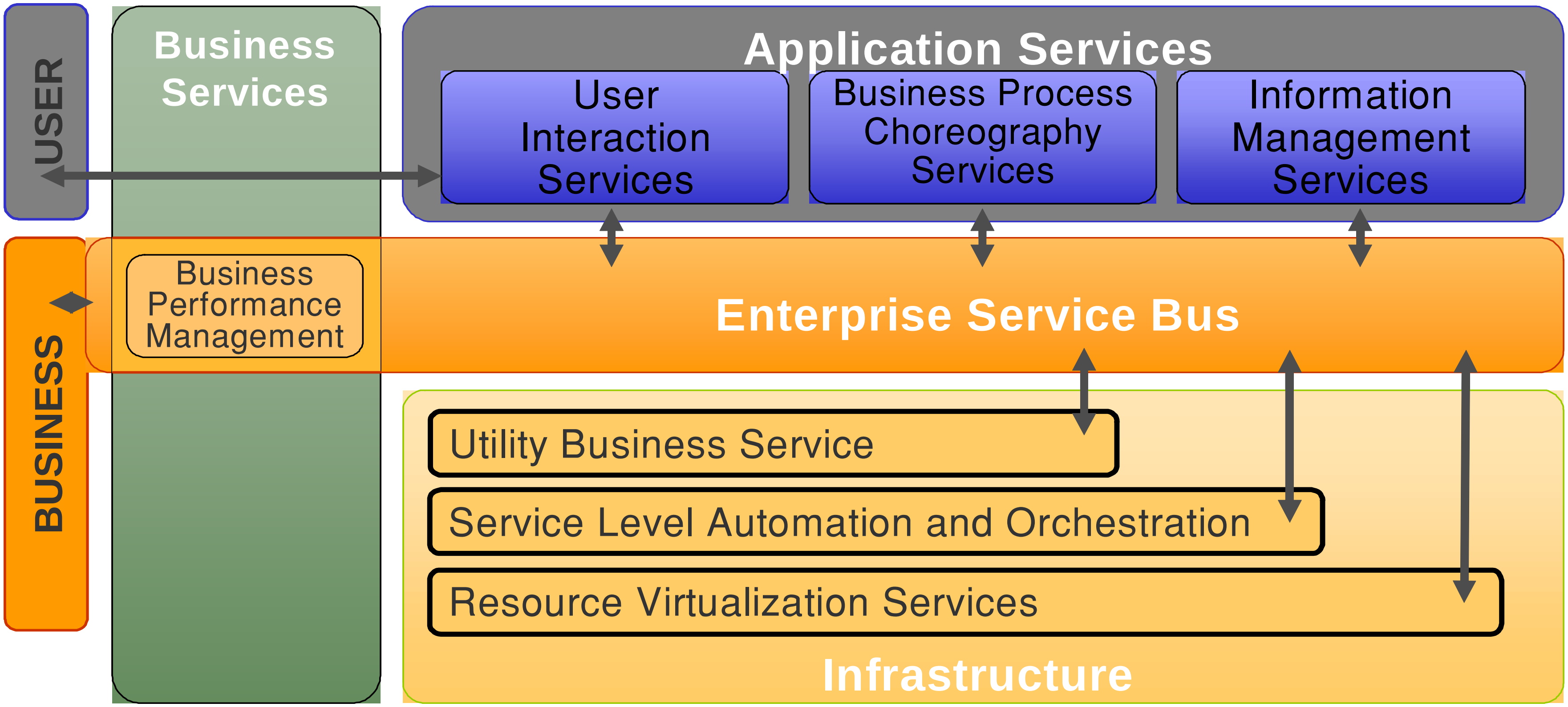


- Flexible
- Variable
- Managed
- Optimized



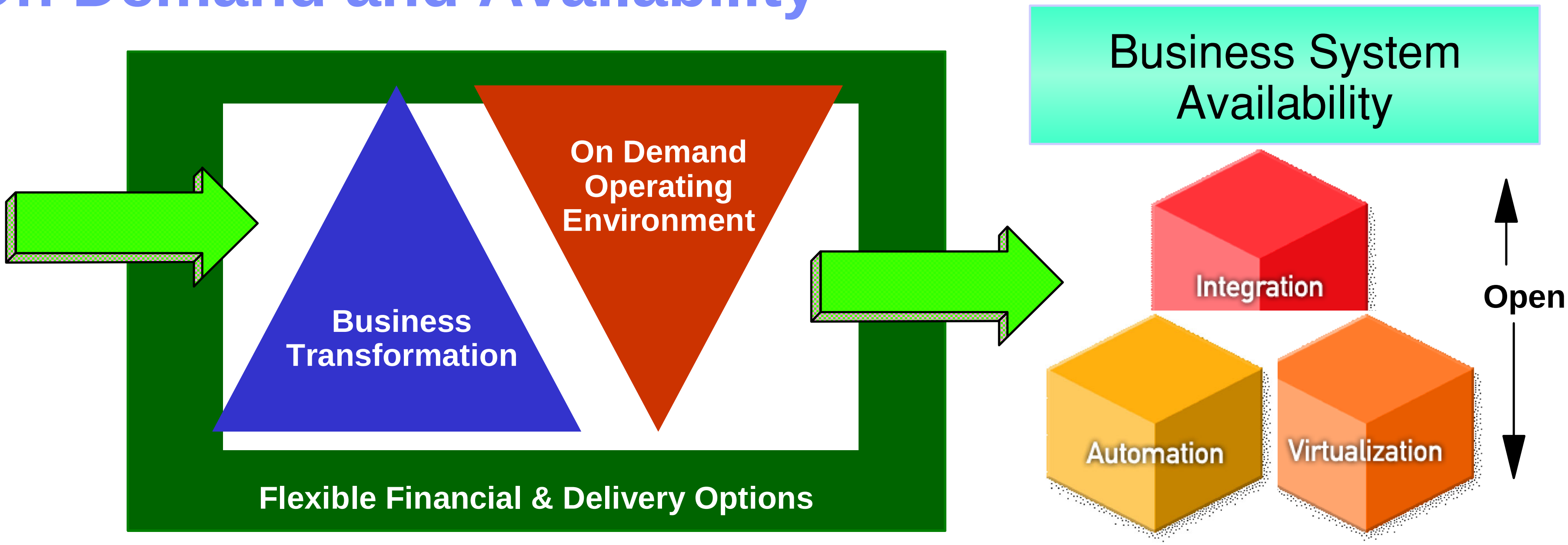
- Open standards and technology to support
  - Integration of people, processes, and information
  - Autonomic capabilities - self configuring, healing, optimizing, protecting
  - Virtualized component usage and management

On Demand Operating Environment


USER
Business Services
Application Services
User Interaction Services
Business Process Choreography Services
Information Management Services
BUSINESS
Business Performance Management
Enterprise Service Bus
Utility Business Service
Service Level Automation and Orchestration
Resource Virtualization Services
Infrastructure

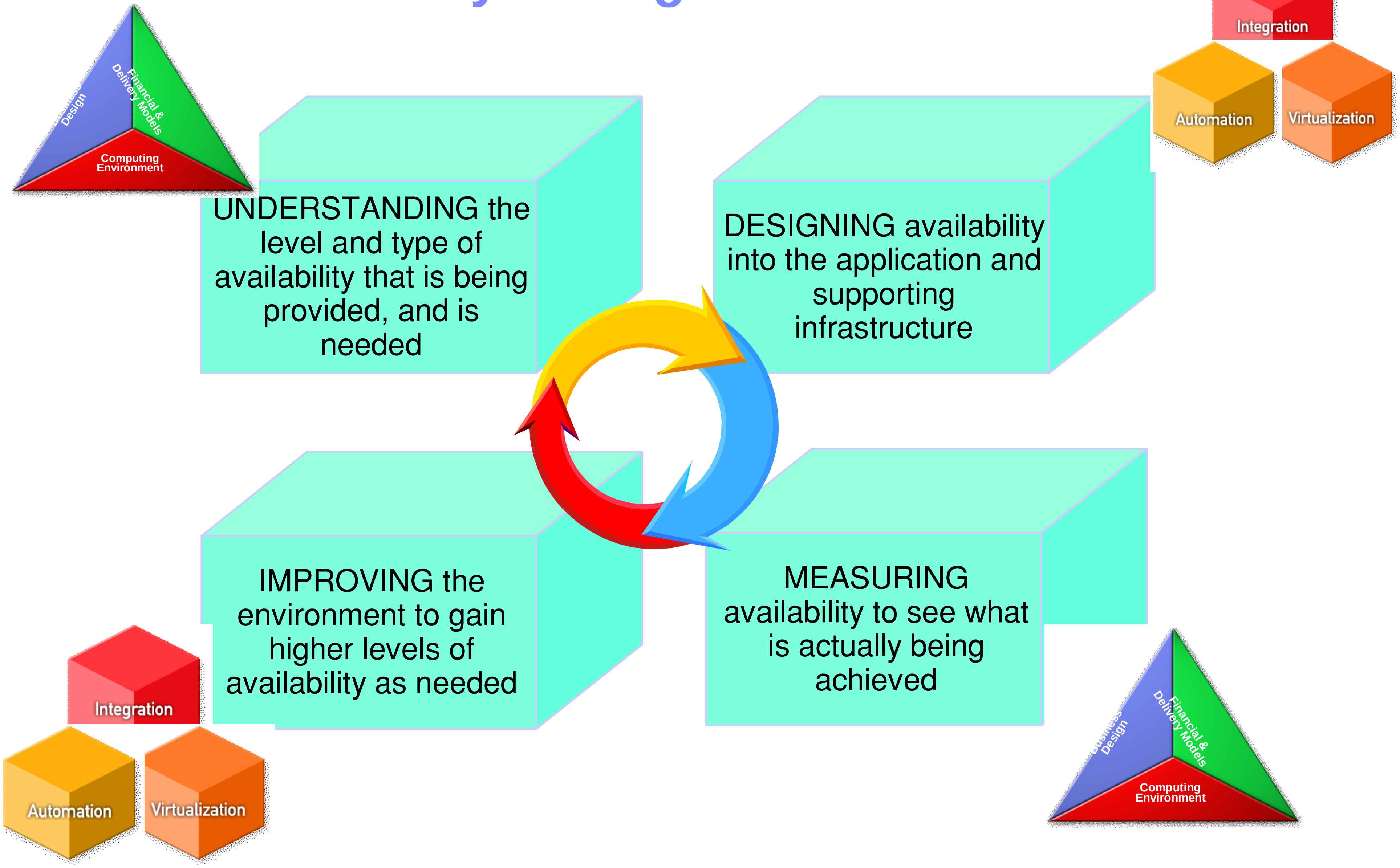
- Integration
  - Business Modeling
  - Process Transformation
  - Application & Information Integration
  - Access
  - Collaboration
  - Business Process Management
- Infrastructure Management
  - Availability
  - Security
  - Optimization
  - Provisioning
  - Policy-based Orchestration
  - Business Service Management
  - Resource Virtualization

On Demand and Availability


On Demand Operating Environment
Business Transformation
Flexible Financial & Delivery Options
Business System Availability
Integration
Automation
Virtualization
Open

- The On Demand business environment requires **On Demand Availability**
- Availability is a key characteristic of The On Demand business environment
- The On Demand Operational Environment must support the availability characteristics of multiple interacting business systems
- Availability efforts are focused at a business, not technology, perspective
  - The first question is not "Is the server up?", but "Is the business process providing service?"
- Implementing and improving availability is done using the operational environment characteristics
- **Managing availability** is required to achieve the desired results

What is Availability Management?



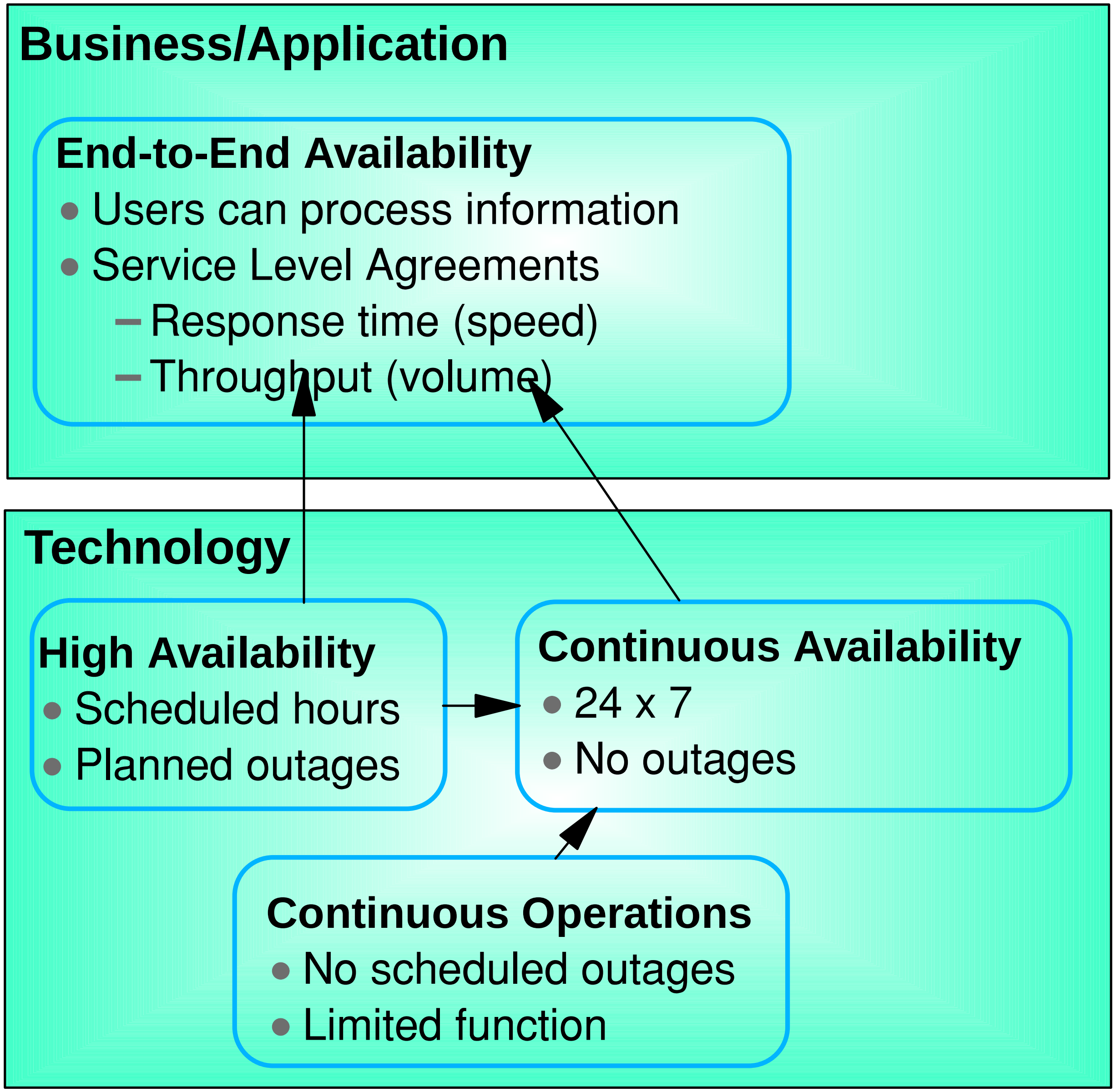
UNDERSTANDING the level and type of availability that is being provided, and is needed

DESIGNING availability into the application and supporting infrastructure

IMPROVING the environment to gain higher levels of availability as needed

MEASURING availability to see what is actually being achieved

Understanding Availability



Business/Application

End-to-End Availability

- Users can process information
- Service Level Agreements
  - Response time (speed)
  - Throughput (volume)

Technology

High Availability

- Scheduled hours
- Planned outages

Continuous Availability

- 24 x 7
- No outages

Continuous Operations

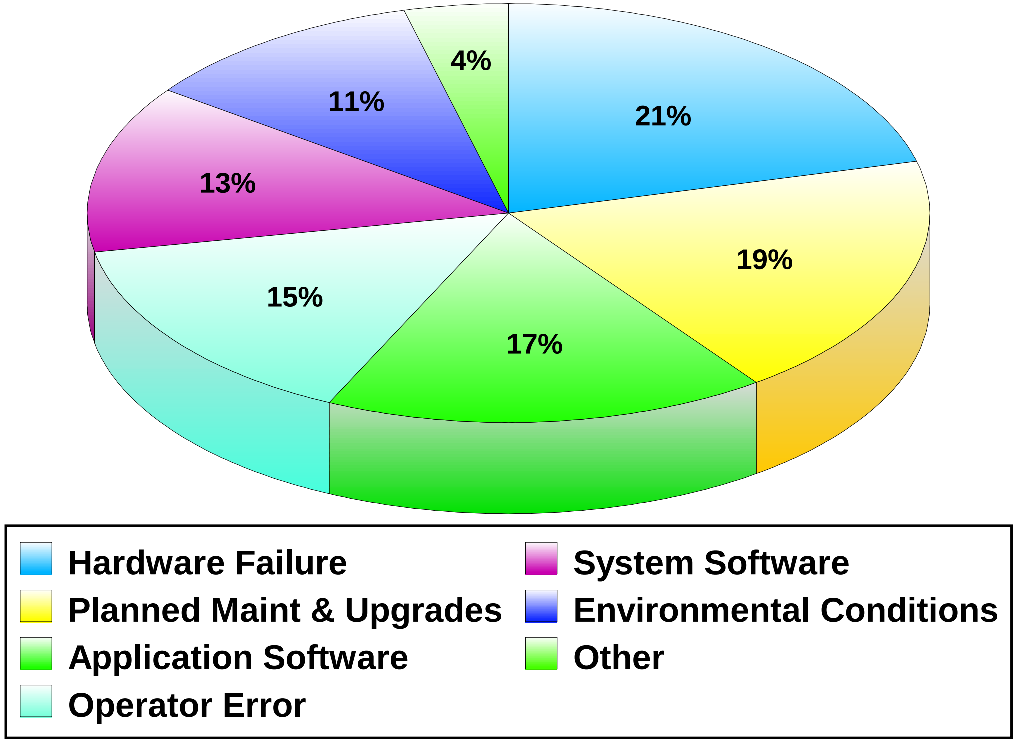
- No scheduled outages
- Limited function

Outages

- Planned
  - Maintenance
  - Upgrades
  - Changes
  - Data ReOrgs
  - Conflicting workloads
  - Facilities
- Unplanned
  - Technology errors
  - Human errors
  - Attacks
  - Natural acts

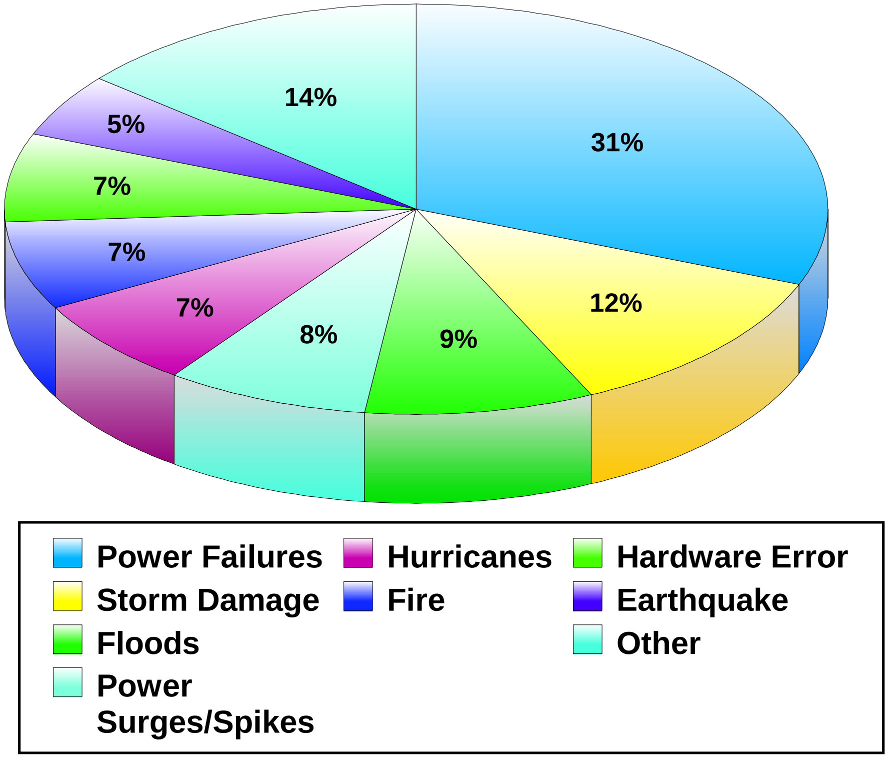
What Impacts Availability?

Corporate Computer Down Incidents



Source: Standish Group Research

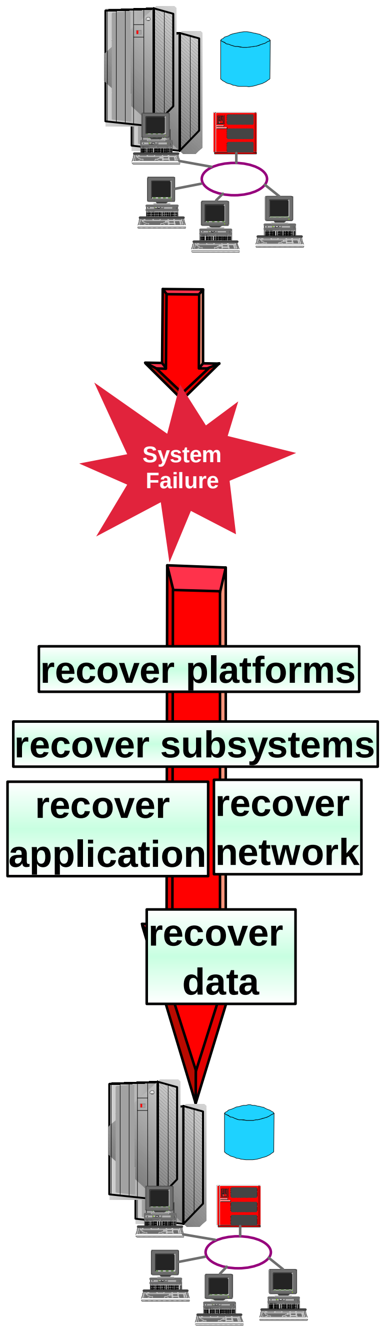
Corporate Computer Disaster Incidents



Source: Contingency Planning Research

Recovering from an Outage

- **Infrastructure**
  - Re-establishing physical and logical connectivity
- **Business Logic**
  - Restarting the application logic
- **Data**
  - Restoring the data to the most consistent state before the outage
  - Database recovery (last good image + logs)
  - Database restart only (last good image, if taken recent enough)
- **Recovery Time Objective (RTO)**
  - Time needed to recover from an outage
  - How long one can afford to be down
- **Recovery Point Objective (RPO)**
  - Amount of data that can be recreated during a recovery
  - Defines tolerance for data loss
- **Network Recovery Objective (NRO)**
  - Time needed to switch over network



System Failure

recover platforms

recover subsystems

recover application

recover network

recover data

What aree Realistic Availability Levels?

"Class of 9's " - Availability Metrics

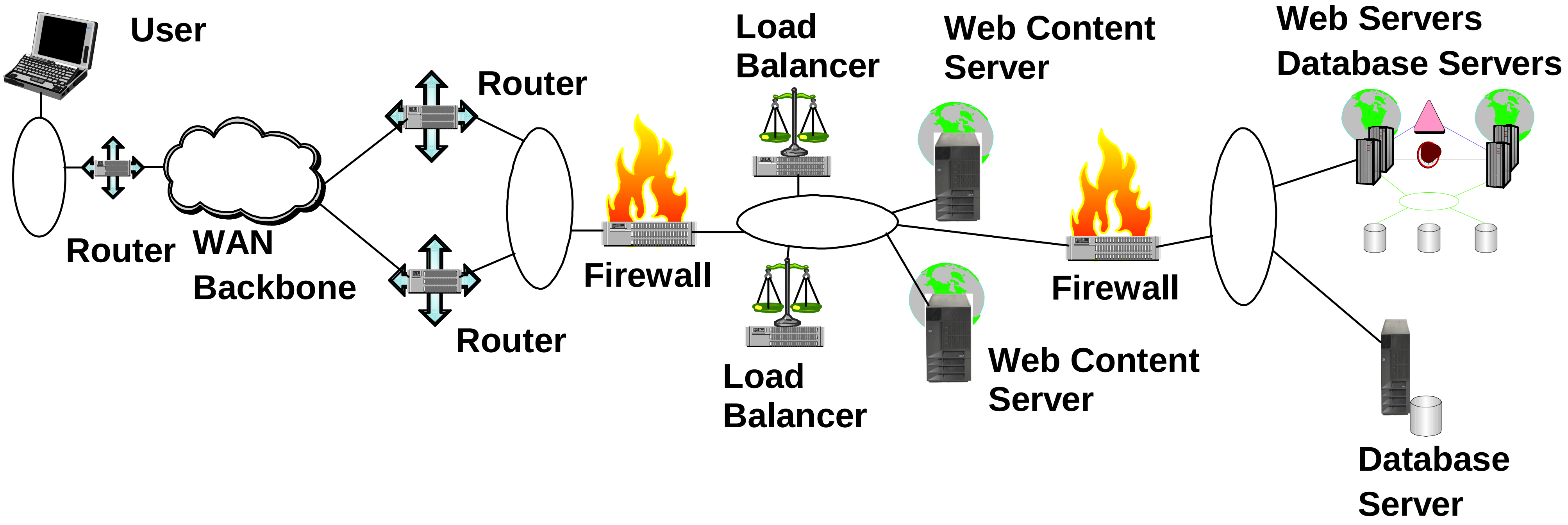
	Availability Percentage	Outage Minutes Per Year (24 x 365)	Outage Cost Per Year ($50,000/hour)
Unmanaged	90.0	52,560	$43,800,000
Managed	99.0	5,256	$4,380,000
Well Managed	99.9	525.6	$438,000
Fault Resilient	99.99	52.6	$43,800
High Availability	99.999	5.3	$4,300

Source: Strategic Research Corp.

- ***These numbers do not take into account:***
  - ***Subsystem/Application Software***
  - ***Network***
  - ***Cross platform applications***
  - ***Site***
- ***Goal: for end-to-end, 97% is minimum, 99% or above is best***

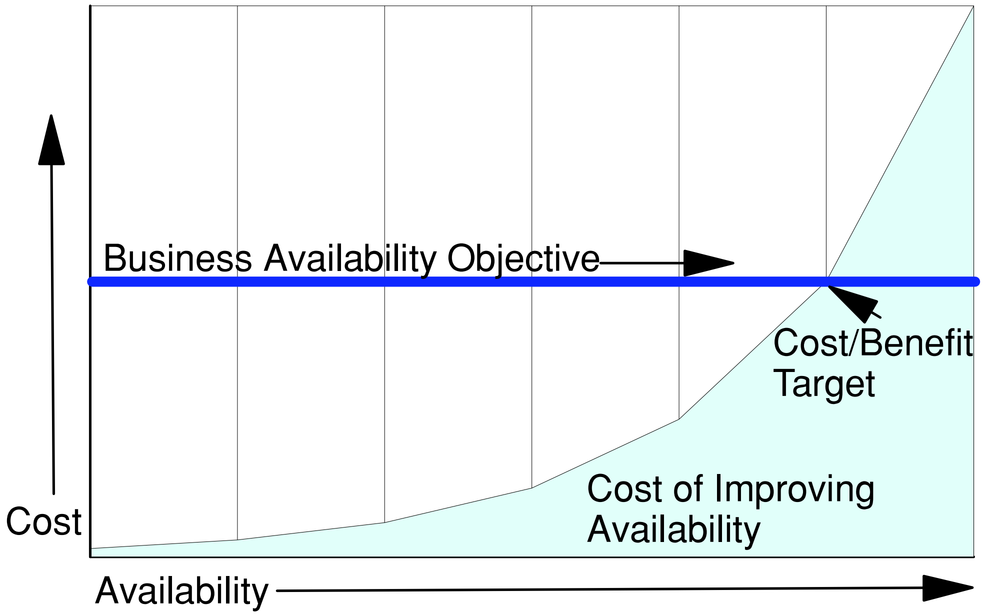
Why Design For Availability?

- There are an increasing number of parts between the users, application components, and data components
- The more parts, the more opportunity for failure, without a design



- 30 components at 99% each = 73.97% end-to-end availability
- 10 at 100%, 20 at 99% =81.79%
- 20 at 100%, 10 at 99% =90.44%

Availability Design Goals



Eliminate Outages (Reactive)

- Reduce FREQUENCY
- Minimize DURATION
- Limit SCOPE

Plan Systems and Applications (Proactive)

- Minimize/eliminate single points of failure and disruptive activities
- Implement fast error detection and recovery actions
- Determine capacity and growth to predict current/future impact on availability, and take efforts to provide it

Approach

- Hardware and software functions are the foundation - "raw capabilities"
- Apply availability improvement techniques to transform these capabilities into solutions
- There will always be a between improvement target and cost

Designing for Availability - using On Demand

Integration

- Effective Systems Management
- Automating Operational Processes

Autonomic

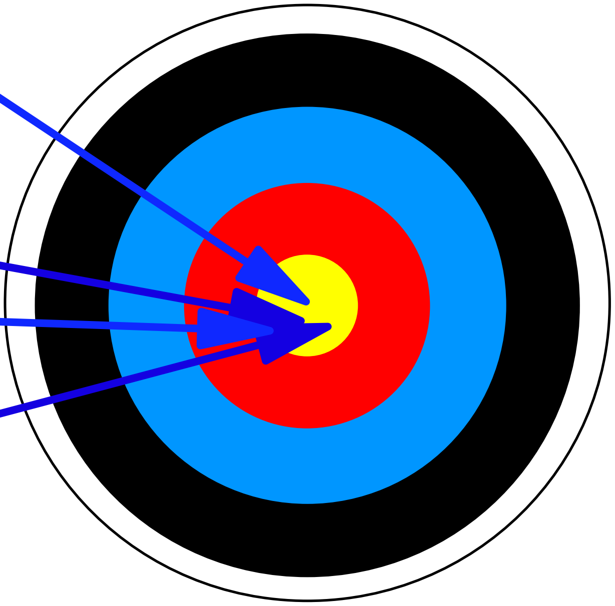
- Automating Operating Processes

Virtualization

- Configuration Options
  - Redundancy
  - Isolation
  - Disaster Recovery and Business Continuity

Standards

- Standards Deployment
- Application Design
- Integrated testing



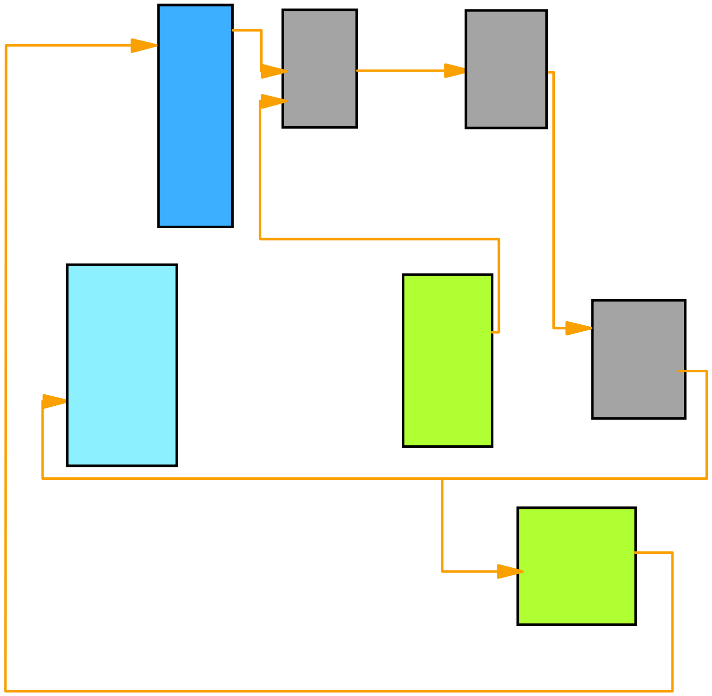
Availability Target

Systems Management

- Improve availability by honing the processes, procedures, policies, skills, and tools inherent to the management of an I/T organization

Elements to analyze

- How does the process (or steps in the process) support the application business requirements
- What process steps are impacting availability
- What is the measured (not just perceived) impact
- What potential improvement could be gained via:
  - Reducing process step length
  - Using products or product functions
  - Efficient data sharing with other processes
  - Organization changes
  - Responsibility changes

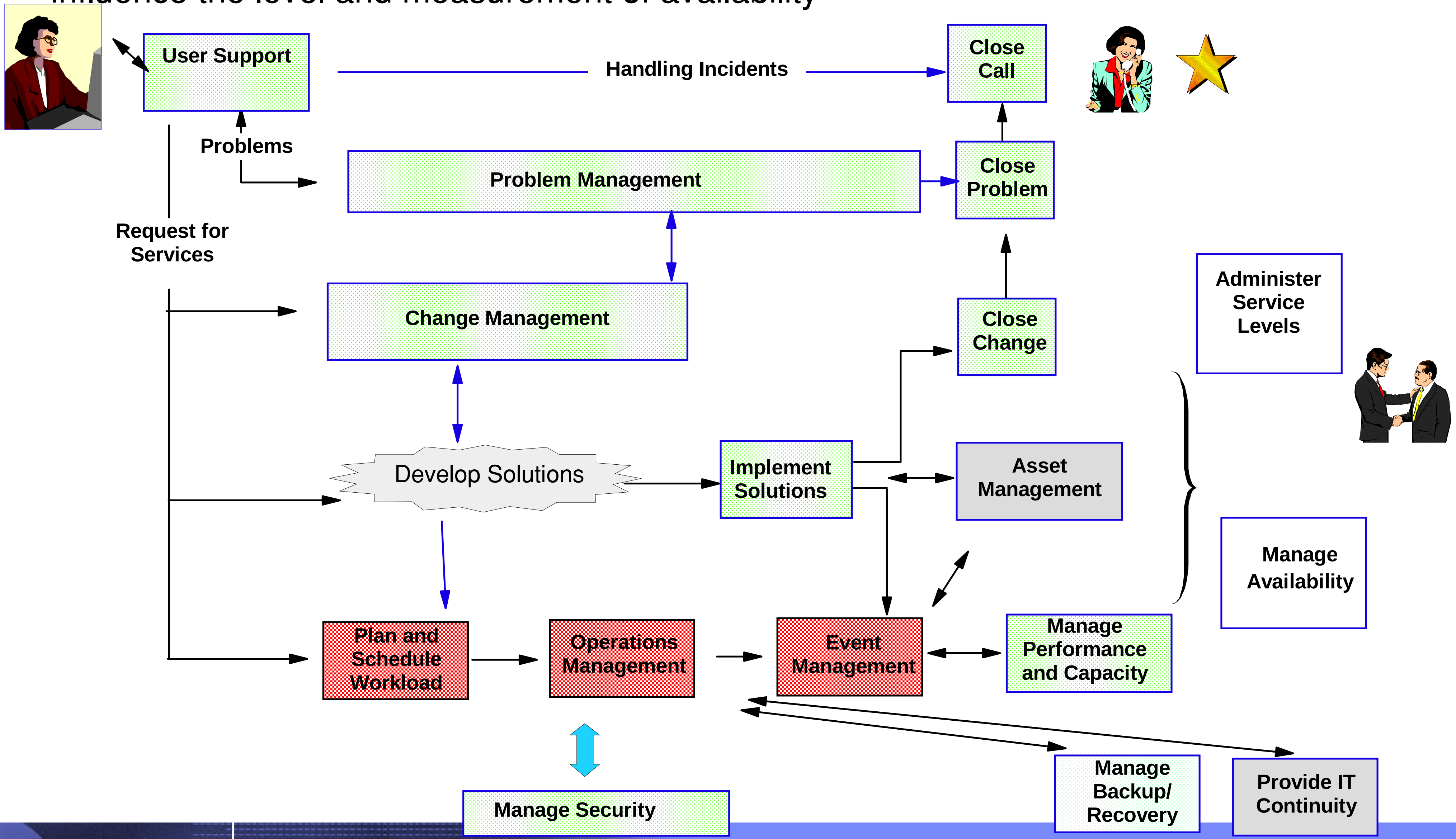


Assessment/improvement frameworks

- Systems Management Framework Design (SMFD)
- Information Technology Process Model (ITPM)
- IT Infrastructure Library (ITIL)

Service Delivery Example

- Systems management processes that integrate to support service delivery can influence the level and measurement of availability



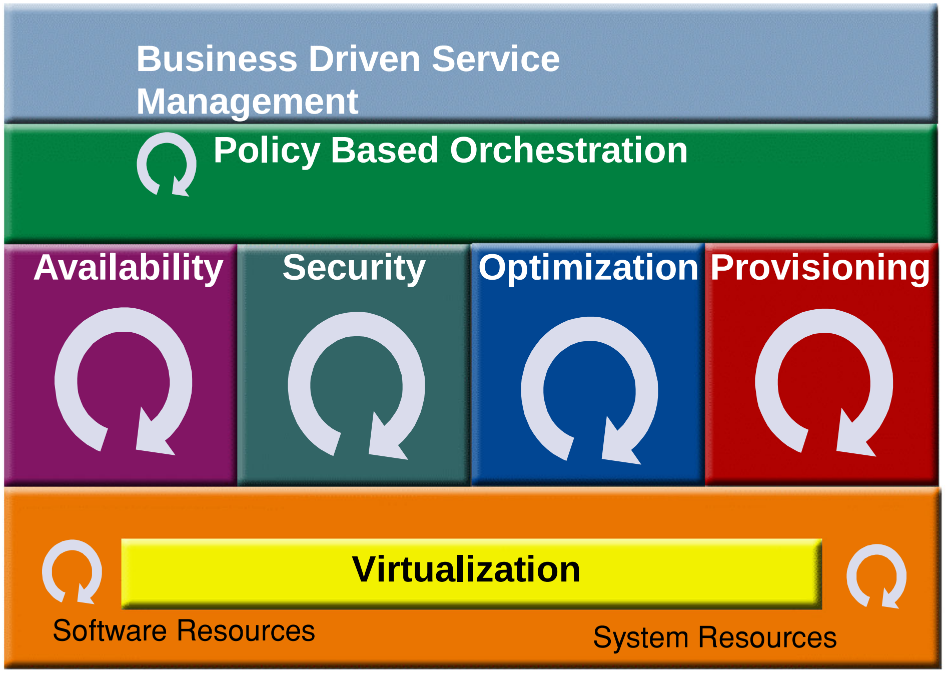
Basic Process Questions

- **Problem Management**
  - What are the root causes of problems?
  - How long are problems taking to resolve? What contributes to this?
- **Change Management**
  - Are impacts of changes to an applications "end-to-end" infrastructure known?
- **Operations Management**
  - Are procedures documented, up to date, and accessible?
  - Are operational tasks (startup, shutdown, monitoring, recovery) carried out as efficiently as possible?
- **Event Management**
  - Can we detect events indicating situations that impact or may impact availability?
  - Are those events being used to notify someone or invoke resolution actions?

Automation

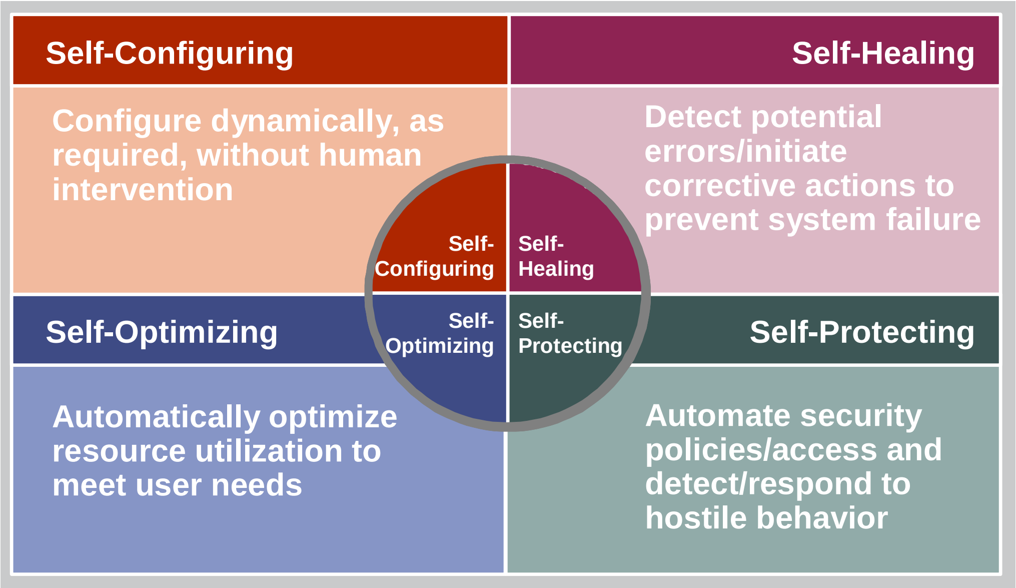
Reduce or eliminate human intervention in the operating environment.

➡ Starts the foundation for moving into On Demand management

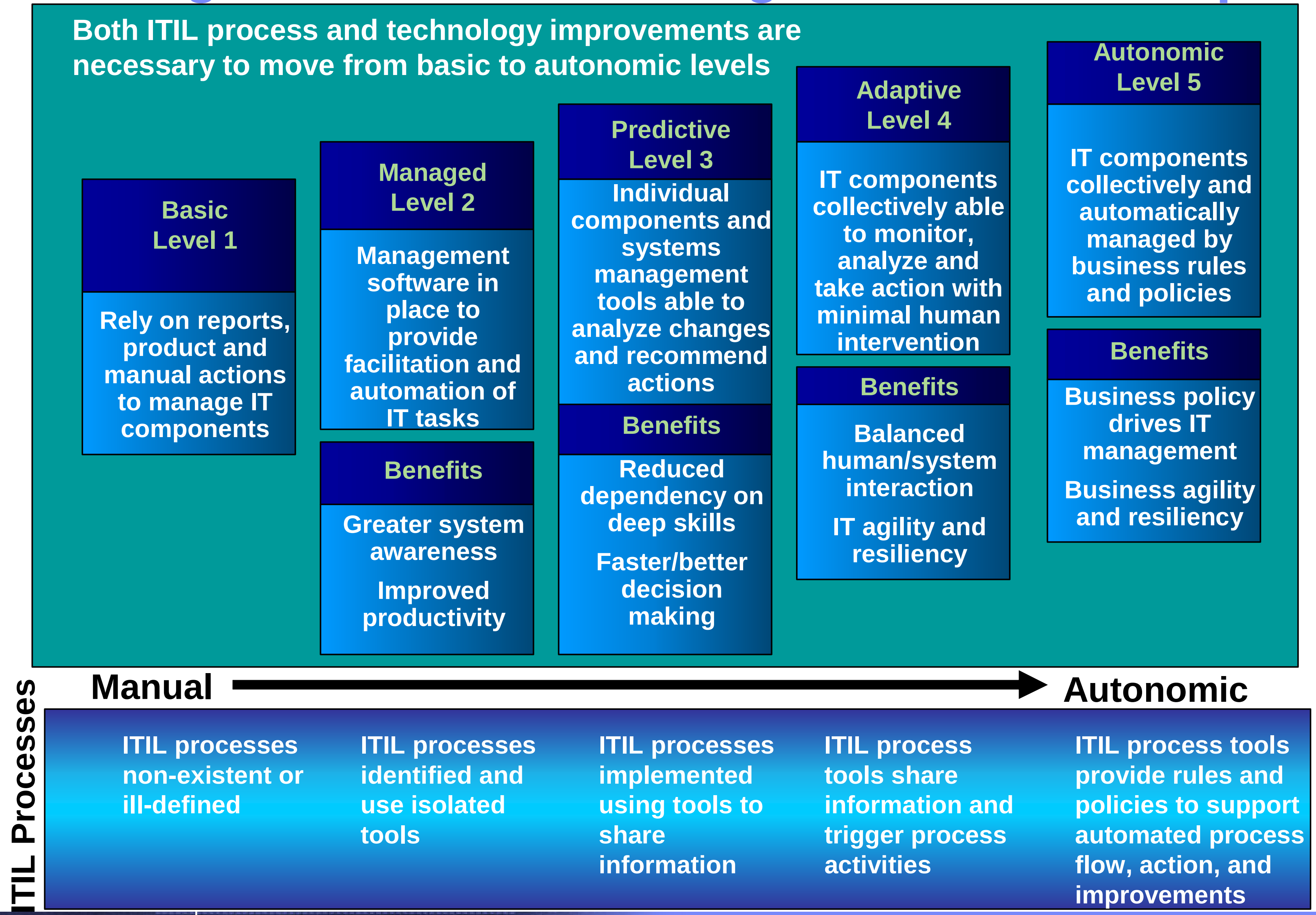


Where to Use

- Systems Management process efficiency
- Operational tasks
  - Scheduling
  - Systems and Data provisioning
  - Distribution (both software and output)
  - Event detection and response
- Monitoring (availability and performance)


Self-Configuring
Self-Healing
Configure dynamically, as required, without human intervention
Detect potential errors/initiate corrective actions to prevent system failure
Self-Configuring
Self-Healing
Self-Optimizing
Self-Optimizing
Self-Protecting
Self-Protecting
Automatically optimize resource utilization to meet user needs
Automate security policies/access and detect/respond to hostile behavior

Evolving ITIL Processes using Autonomic Computing



Both ITIL process and technology improvements are necessary to move from basic to autonomic levels

Basic Level 1

Rely on reports, product and manual actions to manage IT components

Managed Level 2

Management software in place to provide facilitation and automation of IT tasks

Benefits

Greater system awareness

Improved productivity

Predictive Level 3

Individual components and systems management tools able to analyze changes and recommend actions

Benefits

Reduced dependency on deep skills

Faster/better decision making

Adaptive Level 4

IT components collectively able to monitor, analyze and take action with minimal human intervention

Benefits

Balanced human/system interaction

IT agility and resiliency

Autonomic Level 5

IT components collectively and automatically managed by business rules and policies

Benefits

Business policy drives IT management

Business agility and resiliency

Manual → Autonomic

ITIL Processes

ITIL processes non-existent or ill-defined

ITIL processes identified and use isolated tools

ITIL processes implemented using tools to share information

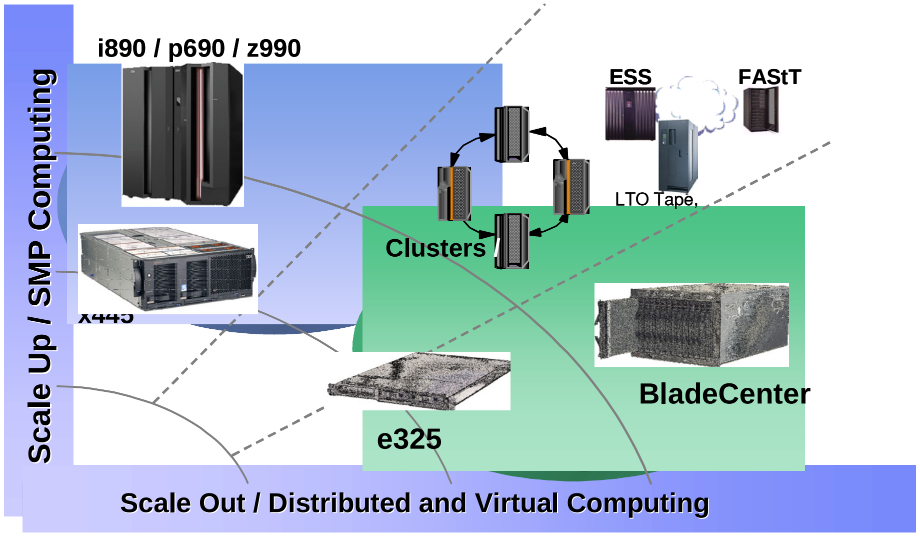
ITIL process tools share information and trigger process activities

ITIL process tools provide rules and policies to support automated process flow, action, and improvements

Redundancy

A key use of On Demand Virtualization

- Hardware platforms
  - Blade Center, LPAR, TotalStorage...
- Application platforms
  - Operating system platforms
  - Parallel Sysplex
  - Operating System Clustering solutions
- Applications and subsystems
  - DB2 data sharing
  - CICS AOR/TOR configuration
  - WebSphere clustering
- Network components and connectivity
  - Multiple communications paths
  - Virtual IP Addressing (VIPA)
  - Load Balancers
- Storage Subsystems and Data
  - RAID, Mirroring, Flash Copy
  - Peer-to-Peer Remote Copy (PPRC)
  - Extended Remote Copy (XRC)



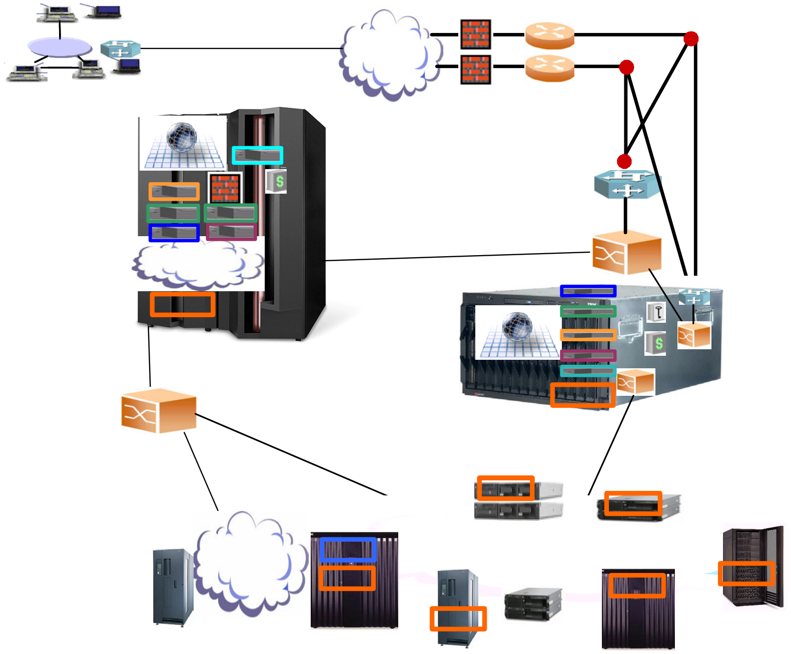
- Masks individual component outages
- Little or no end user availability impact, depending on redundancy level
- Reduces the application outage scope and/or duration
- Increases flexibility for allocating resources to ensure availability

Redundancy Design

- **Backup configuration - 1 to 1, or 1 to N?**
  - 1 to N is cheaper to implement - but more exposure if multiple components fail
  - 1 to 1 is more expensive, but reduces exposure to multiple component failure
    - A virtualized backup environment can reduce 1 to 1 costs
- **What is the required Implementation Level?**
  - Cold -> Warm Standby ->Hot Standby ->Fault Tolerant
- **Process for normal->failover->return to normal?**
  - Detection of potential problem
  - Actions to move to backup environment (can they be automated?)
  - Actions to return to normal environment (can they be automated?)
- **Ability of system and application software to run in a failover environment?**
  - Operating system, subsystem, and application definitions - what has to change, and how?
- **Mirrored data**
  - What is appropriate based on Recovery Point Objective?
- **Management**
  - Can monitoring/control be done at both the "virtual" and individual component levels?

Isolation

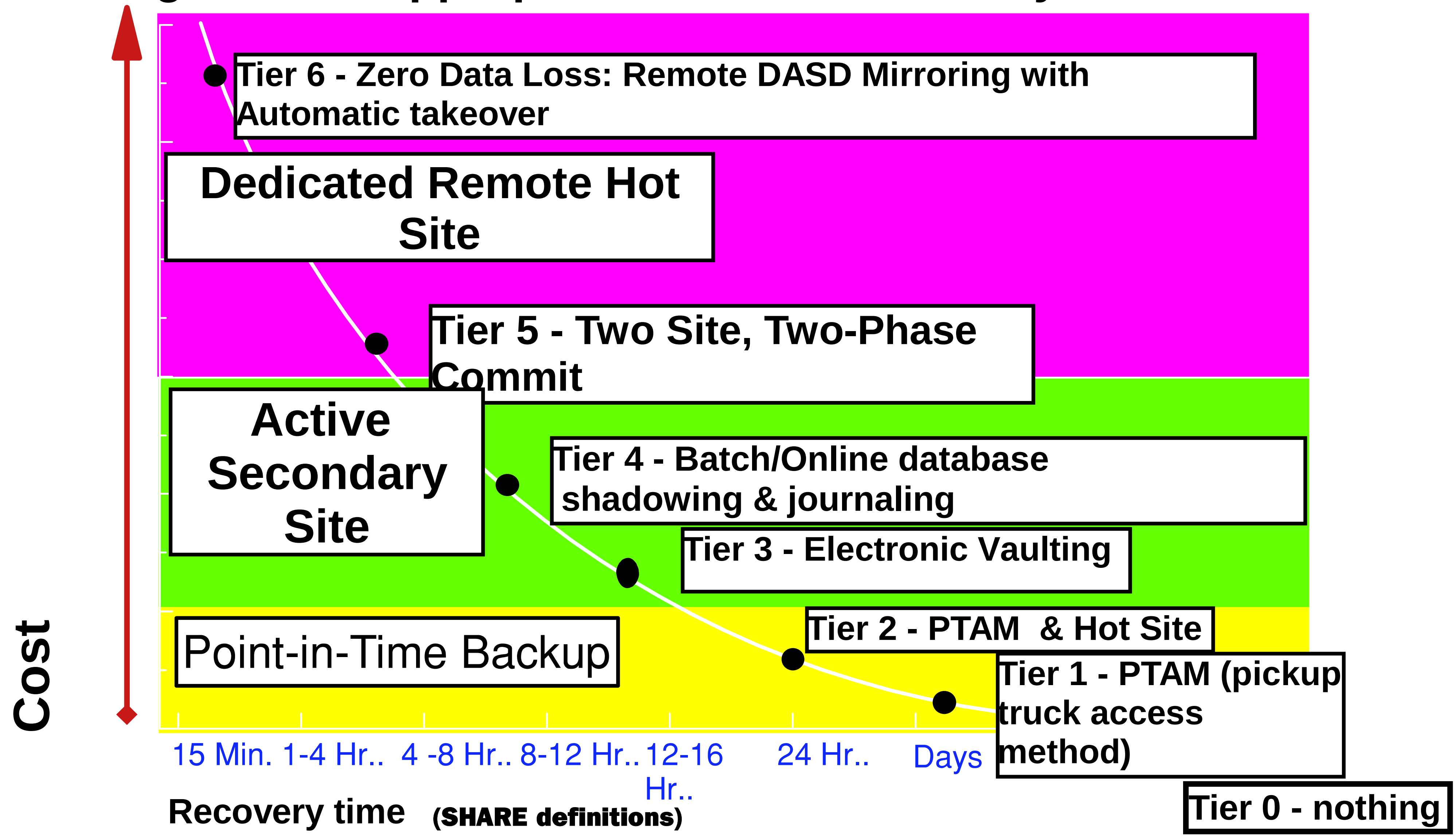
- Physical or logical separation
- Avoid conflicts between workloads and resource usage
- Avoid potential change and performance impacts
  - Grouping workloads to assign access to computing resources
  - Applying "quality of service" techniques to isolate bandwidth
  - Separating I/O intensive and compute intensive workloads
  - Separating test and production workloads
- On Demand virtualization can leverage isolation and reduce cost
  - Consolidation of different Operating Systems on fewer hardware platforms
  - Logical storage isolation within physical devices
  - Network bandwidth QoS for network traffic



- Minimize availability exposures due to changes
- Less resource contention for critical resources
- More controllable function migration
- Limit costs by applying improvements to most critical functions

Business Continuity/Disaster Recovery

➡ Design for the appropriate Disaster Recovery tier



Tier 6 - Zero Data Loss: Remote DASD Mirroring with Automatic takeover

Dedicated Remote Hot Site

Tier 5 - Two Site, Two-Phase Commit

Active Secondary Site

Tier 4 - Batch/Online database shadowing & journaling

Tier 3 - Electronic Vaulting

Tier 2 - PTAM & Hot Site

Point-in-Time Backup

Tier 1 - PTAM (pickup truck access method)

Tier 0 - nothing

Cost

15 Min. 1-4 Hr.. 4 -8 Hr.. 8-12 Hr.. 12-16 Hr.. 24 Hr.. Days

Recovery time (SHARE definitions)

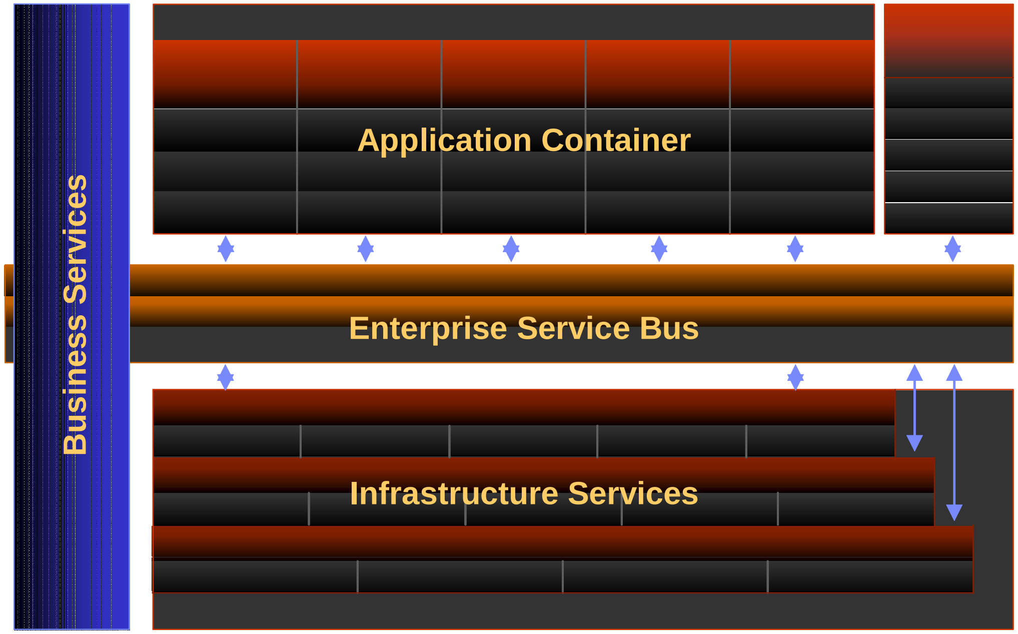
Recovery Tiers

Tier / Description	Recovery Point Objective (RPO)	Recovery Time Objective (RTO)	Enterprise Percentage
0 / No D/R plan	-	-	< .3 %
1 / PTAM	24-48 H	> 48 H	< .1 %
2 / PTAM and hot site	24-48 H	24 H	90 %
3 / Electronic vaulting	< 24 H	< 24 H	6 %
4 / Active 2nd site	seconds	< 24 (< 2) H	< .5 %
5 / 2nd site, 2 phase commit	seconds	< 2 H	< .1 %
6 / Zero data loss	none/ seconds	< 2 H	3 %

- Geographically dispersed IT facilities
- Data backup planning and execution is critical
- Disaster recovery readiness of critical suppliers, vendors

Standards

➡ Improve availability by reducing system complexity via consistent definitions and policies



Examples

- Use of industry/published standards for integration functions
- Naming conventions for component grouping/identification
- Configuration information to easily identify platform or connectivity information
- System and application definitions to permit automated provisioning and cloning

Applicable to all areas

- Required for On Demand policy based orchestration
- Foundation for Services Oriented Architecture

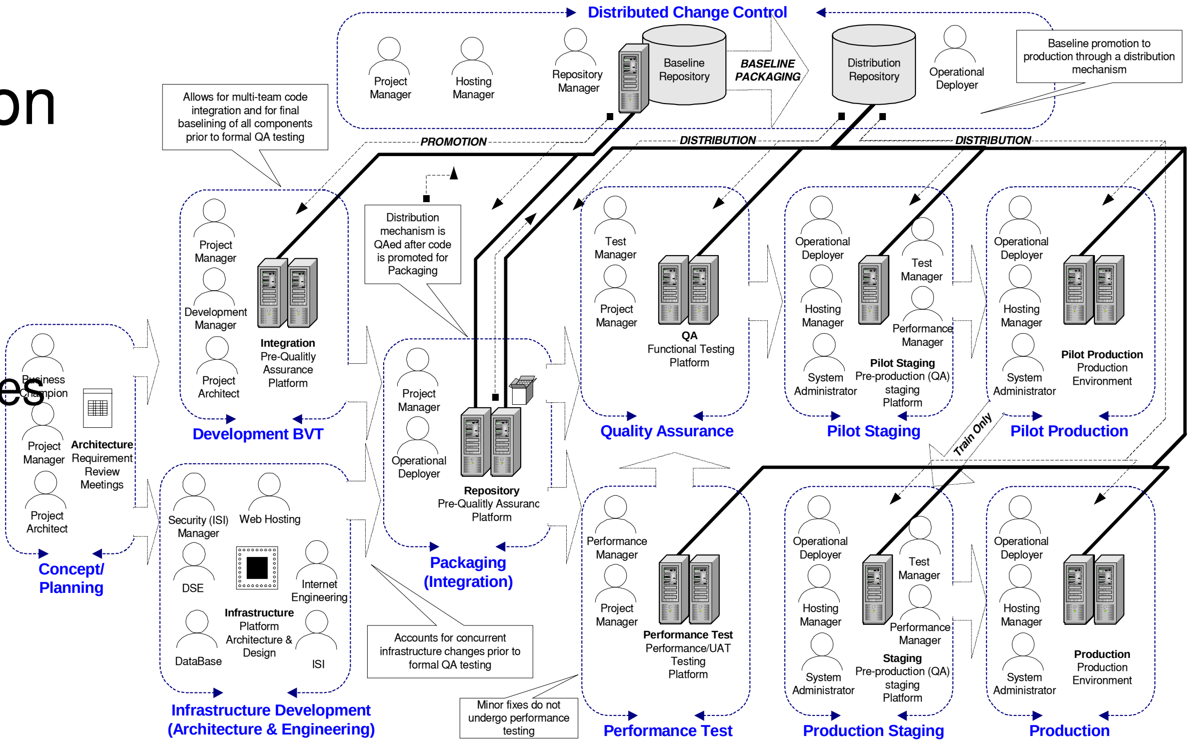
Benefits

- Better utilization of design techniques
- Consistent policy administration and operations
- Increased stability
- Easier to implement automated operational scenarios
- More efficient training and skills usage

Application Design

Objective: Improve availability by designing applications that exploit technology availability features

- Best done early in application development lifecycle
- Examples
  - Eliminate operator intervention
  - Use standard, documented interfaces
  - Use subsystem availability features
  - No designed outages
  - Fast restarts
- Recommendations
  - Involve users
  - Establish design guidelines and standards
  - Ensure compliance during "Design Review" phase of the project
  - Cooperation between Application Development and Service Delivery Processes



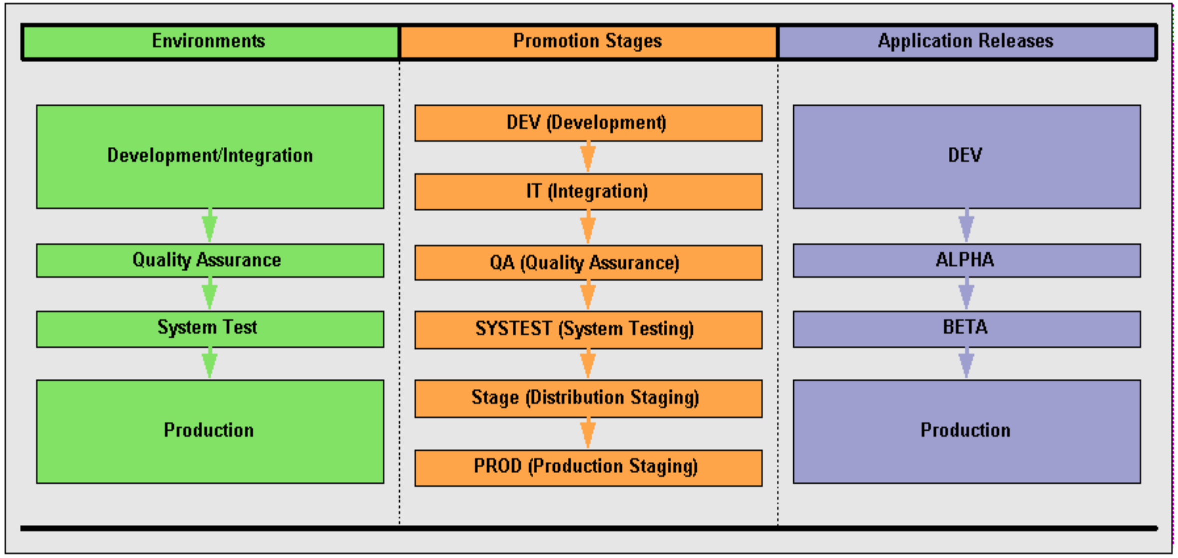
Benefits

- Availability addressed "at the source"
- Less retrofitting of availability design
- Greater awareness of availability beyond systems and operations group

Integrated Testing

Objective: Improve availability by creating and maintaining a testing environment that crosses all components

- "End-to-end" infrastructure and application testing
- Availability and performance
- Repeatable, Controlled, Targeted and Automated testing
- Pre-production quality/acceptance
  - Availability and performance
  - Automation and recovery
- Organization responsibilities
  - Independent
  - Power to define and enforce criteria



Environments
- Development/Integration
- Quality Assurance
- System Test
- Production

Promotion Stages
- DEV (Development)
- IT (Integration)
- QA (Quality Assurance)
- SYSTEST (System Testing)
- Stage (Distribution Staging)
- PROD (Production Staging)

Application Releases
- DEV
- ALPHA
- BETA
- Production

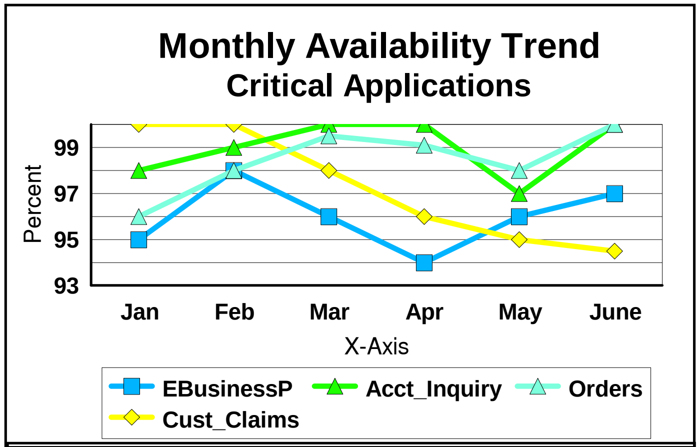
Benefits

- Smoother hardware and software migration
- Improved understanding of reliability and availability before production
- Better support for change process
- Greater confidence in expected results
- Increased end user satisfaction

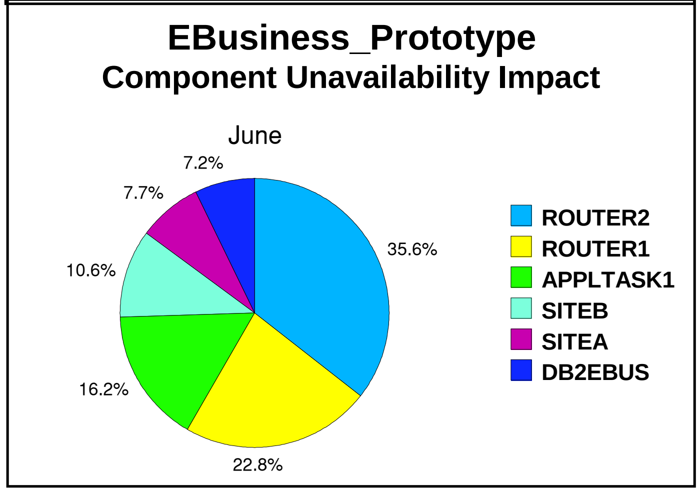
Validating the design - Availability Measurement

"One cannot manage what one is not measuring"

- What is the cost of an outage?
  - Needed to quantify value of availability
  - Will vary by application and over time
  - Avoid "analysis paralysis"
    - start, and adjust as needed
- What should be measured?
  - More than just "percent available"
    - Incident frequency
    - Affected users
    - Lost time (application, component, user)
    - Lost or delayed transactions or workload
    - Outage causes
  - Relate to service levels to determine business impact
- How should it be measured?
  - End-to-end (the application view)
  - Derive from data produced by components or the problem management process
  - Use application platform, data, network, and user measurement points



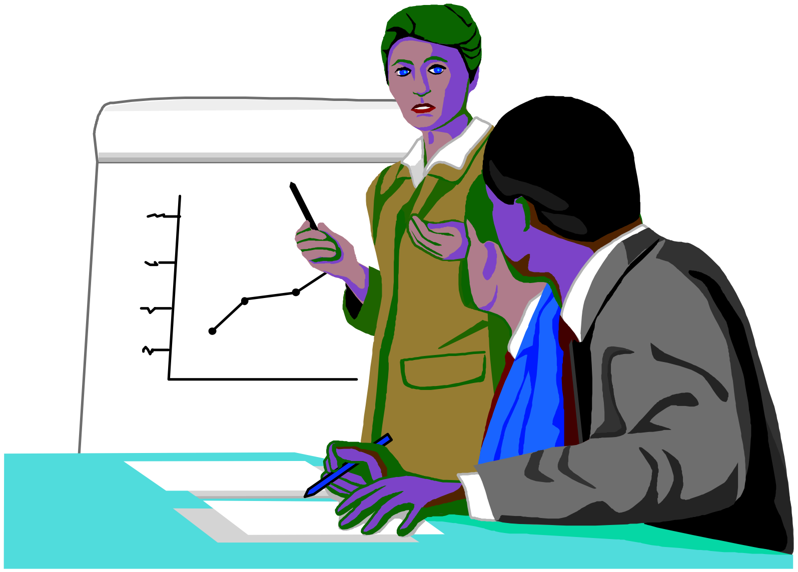
Monthly Availability Trend
Critical Applications



EBusiness_Prototype
Component Unavailability Impact

Measurement Process

- Collect and analyze data from:
  - System and network protocols
  - Monitoring techniques
  - Products
  - Component or management APIs
- Report on
  - Application availability (end-to-end)
  - Component availability (as related to end-to-end availability)
  - Impact (cost) of unavailability
  - Root cause outage categories
  - Relationship to problem, change, performance data

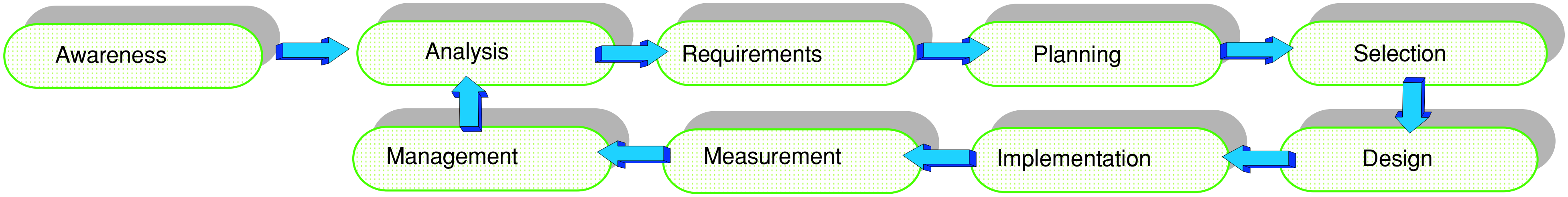


➡ ***Use this information to identify and implement further improvements***

Planning for Availability

- Concentrate on the most important business systems/applications
- Define and document availability requirements
  - Service level agreements (availability and response time requirements)
  - Outage impact/costs
  - Operational dependencies
- Determine and analyze the infrastructure to find exposures
  - Outage analysis for each key component/component group
  - Identification of single points of failure (but those do not exist anymore, right?)
  - Component failure impact analysis (CFIA) review
- Identify availability design alternatives based on the improvement techniques
  - Will it reduce outage frequency, length, scope?
  - Will it eliminate certain types of outages from occurring?
- Evaluate the alternatives relative to the requirements
  - Cost or risk vs. benefit value
  - Timeliness (how quickly can it be done)
  - Additional hardware and/or software
  - New/changed organizational roles and/or skills
  - Migration/conversion costs
  - Other

Summary



- Availability Management is a continuous cycle, requiring:
  - Sound **planning** for and **analysis** of what is required
  - **Effective** systems management
  - **Exploiting** products with **availability** features
  - Carrying out **design, implementation, measurement,** and **management** activities
  - Using **Automation** where possible for speed and consistency
- Managing availability must be done to move towards an On Demand operating environment, in support of On Demand business functions
- Focus on the applications-components relationship to determine how to improve the business through higher availability
- Higher availability can be achieved in an **evolutionary** manner

INFORMATION SOURCES

- Continuous Availability Systems Design Guide (SG24-2085)
- Parallel Sysplex Continuous Availability Guide (SG24-4503)
- IBM TotalStorage Solutions for Disaster Recovery (SG24-6547)
- Enabling High Availability e-business on IBM eServer zSeries (SG24-6850)
- On Demand Operating Environment: Managing the Infrastructure (SG24-6634)
- IBM Redbooks (www.redbooks.ibm.com
  - Many more redbooks with availability items for specific technologies
- IBM High Availability Services
  - http://www.ibm.com/services/its/us/availability.html
- Availability focused/content sites - examples
  - www.availability.com
  - www.itpapers.com
  - www.nextslm.org
  - www.uptimeinstitute.com

RETURN TO INDEX