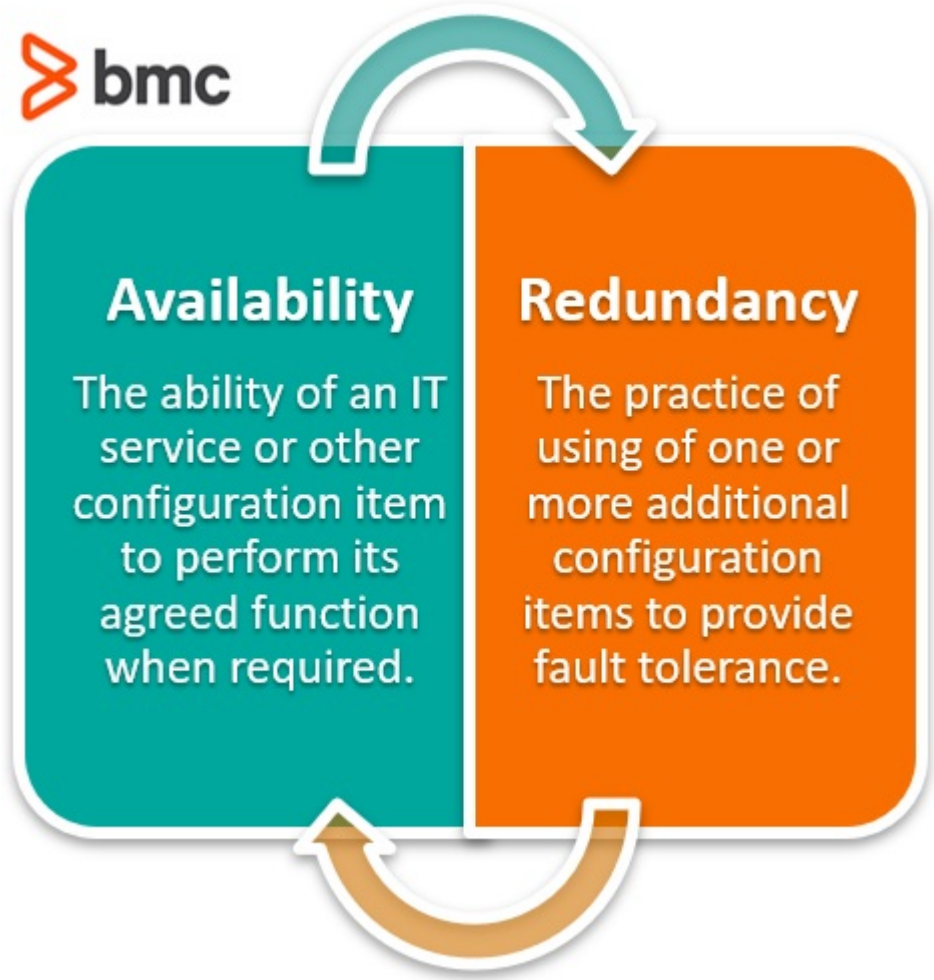
IMPACT OF REDUNDANCY ON AVAILABILITY

When it comes to measuring customer satisfaction, there is no greater determinant than availability of services. Downtime brings disappointment, whether one is filing a tax return, shopping on eBay, or building on Fortnite, as the disruption brings frustration and a loss of benefits and productivity—both personal and business.

Recent research from Infrascale revealed that nearly a quarter of small and medium businesses had their IT systems go offline in 2019, with an estimated 37% losing customers and 17% losing revenue.

To ensure that service availability meets the requirements of customers, it is crucial that the design of the service not only eliminates or minimizes the effect of downtime, but also reinstates the service and component functionality as quickly as possible when disruption occurs. This is the intersection of redundancy and availability.

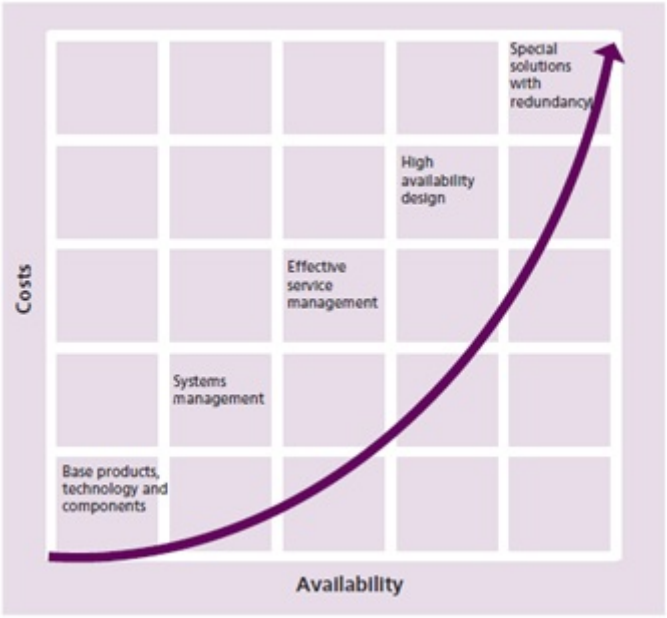


What is availability?

The ITIL® 4 framework defines availability as "the ability of an IT service or other configuration item to perform its agreed function when required." This sounds simple in practice but becomes quite complicated to fully calculate it when considering dependencies such as:

- Service architecture
- Third party and customer components
- Capacity and reliability
- Other parameters

The traditional measure of availability is percentage, with 99.999% seen as the ultimate goal—that's less than 5 minutes of downtime annually. In reality, this is far from achievable. The higher the availability target, the more the cost of not just design of availability, but also the *management* of availability, as shown in the chart below:



Relationship between levels of availability and overall costs (Source)

What is redundancy?

Redundancy is a technique of using of one or more additional configuration items to provide fault tolerance. For example, having applications running on multiple virtual machines hosted on different servers.

Redundancy is all about ensuring that if one service component fails, there are others working in parallel to limit disruption. This involves identifying single points of failure in the entire service architecture and providing additional components that can continue functioning in the event of one failing.

How to implement redundancy

As shown in the image above, implementing redundancy to support availability adds significant costs to business operations, in terms of both components and management. Hence the journey for implementing redundancy must be balanced from a cost-benefit analysis.

The first step is primarily a **business impact analysis exercise** that assesses the impact of disruption to services and determines how much an organization is willing to bear as a loss, which will lead to a decision of how much it is willing to invest to limit the particular loss. A key metric for this decision is the recovery time objective (RTO) which is the period of time following an incident within which a service must be resumed or resources must be recovered.

Then a **review of architecture** follows to determine which components need to work in parallel, and the framework for managing and supporting these components. Based on the investment by the organization, acquisition and integration of these components is carried out. Consideration must also be given to include redundancy in IT environments such as:

- Utilities, e.g. power and cooling
- Connectivity, e.g. multiple internet connections from different providers

For example, most public cloud providers will advertise availability zones which are geographically isolated data centers with independent power sources, networking, and cooling resources. Deploying your application instances in multiple availability zones goes a long way in ensuring high availability of services, by ensuring that downtime to components in one availability zone do not result in total service downtime as other zones continue to serve the customer.

Redundancy in the cloud

In cloud environments, auto-healing is another technique that can be applied in implementing redundancy. This works hand in hand with auto-scaling (automatic distribution of workloads across multiple computing resources such as containers or virtual machines) to ensure a set number of instances are maintained even when disruption occurs.

For example, a rule can be set that a particular application instance always requires four virtual machines running at any one time. So, if the hosting environment detects that one VM is down or unresponsive, it immediately spins a replica VM based on previously configured parameters. This ensures minimal downtime and lessens the need for manual intervention for restoring availability.

Measuring the Impact of Redundancy on Availability

The widely accepted computation for availability is:

$$Availability = \frac{Agreed\ service\ time - Downtime}{Agreed\ service\ time}$$

While this looks simple enough, it is still a challenge to determine agreement and dependencies, as mentioned earlier. Most service providers will use metrics such as MTBF (mean time between failures) and MTRS (mean time to restore service) to characterize availability. For any service, if the frequency of failures is low or speed of restoration is high, in relation to agreed targets, then it is highly likely to satisfy customer requirements. There is a direct relationship between MTRS and RTO.

MTBF will be a direct beneficiary of redundancy, as having multiple components might increase the period between actual service failures that impact a customer. However, the design of the service components must limit dependency, otherwise if disruption affects all the different components, then MTRS will be longer, as those involved in restoring services will need to work on all the replicated components.

Additional resources

For more on this topic, browse our BMC IT Service Management Blog or check out these articles:

- Reliability vs Availability: What's the Difference?
- Resiliency vs Redundancy: What's the Difference?
- System Reliability and Availability Calculations
- Availability Management and the Role of the Availability Manager