6.2 Resiliency

Q Define

Resiliency is the capacity to rapidly adapt and respond to risks as well as opportunities. This maintains continuous business operations that support growth and operate in potentially adverse conditions. The reach and range step of the assessment process examines business-driven, data-driven, and event-driven risks. The objective here is to explore the risk situation for the company, processes, people, or whatever that affects the business of that organization. This can be divided further down after thorough investigation because risk in one section will not be same as risk in another section.

When you use the resilience framework to look at different parts of the company, you are trying to understand whether you have a risk that you can accept, or whether you have a risk that you want to avoid and mitigate. In other words, you may choose to do nothing about a risk, or you may improve your infrastructure to help ensure that you can handle events if they occur.

You may also decide that the risk is one that you would prefer to transfer to somebody else, such as business continuity and resiliency services. A lot of organizations feel more comfortable transferring risks associated with business continuity to cloud vendors rather than handling risks themselves, as recovery centers are designed to be robust and ensure resilience in the face of a disruption. Additionally, transferring the risk can be accomplished through managed security or resiliency services. This allows you to concentrate on strategic initiatives and leaves the day-to-day management and monitoring of your availability and security configurations to staff locations. So, what can we recommend to create a framework of resiliency? The resiliency blueprint includes different layers – facilities, technology, applications and data processes (both IT and business), organization, and finally, strategy and vision.

The framework enables us to examine the business, understand what areas of vulnerability you might have come across – business-driven, data-driven, and event-driven risks – and quickly pinpoint areas of concern and help you understand what actions you can take to reduce the risks associated with those areas.

6.2.1 Resiliency Capabilities

Q. all capa

The strategy combines multiple parts to mitigate risks and improve business resilience in the following manner:

1. From a facilities perspective, you may want to implement power protection.
2. From a security perspective, that is, to protect your applications and data, you may want to implement a biometrics solution. You might want to implement mirroring, remote backup, identity management, e-mail filtering, or e-mail archiving.
3. From a process perspective, you may implement identification and documentation of your most critical business processes. You may split the functions of processes. You may also want to implement specific requirements confirming to government regulations and standards.
4. From an organizational perspective, you may want to take an approach that addresses the geographic diversity and backup of workstation data. You may want to implement a virtual workplace environment.
5. From a strategy and vision perspective, you would want to look at the kind of crisis-management process you should have in place. You may also want to examine how you can clearly articulate your security policies to everybody and how you implement change management.

agility (e.g.,

6.3 Provisioning

9 Definition provisioning

The provisioning process is a service that uses a group of compliant processes called 'Solution Realization'. Environment provisioning roles separate the preparation tasks and assurance tasks from provisioning tasks. Provisioning design decouples provisioning build and integration activities from requirements, design, procurement, and hardware setup. The process formalizes Quality Assurance (QA) testing in preparation for turning over the provisioned product to the customer. Provisioning is a broad-based service that begins with a Request for Service (RfS) to build a fully provisioned environment for the purpose of hosting an application and database. Provisioning can also be invoked when a major modification must be made to the existing environment. Provisioned environments include development, test, QA, production, and DR. Provisioning defines and communicates the information that is required to begin provisioning. The output from provisioning is an environment configured and tested with an appropriate hardware platform, storage, network, operating system, middleware, other system software, backup capability, monitoring capability, and with the application installed per requirement. In this way

infrastructure required

6.3.3 Benefits

This section discusses the benefits of provisioning.

1. **Ability to measure progress of all the work related to one RfC:** It supports the ability to deliver to service levels.
2. **Continuous improvement activities based on process measurements:** It enables eliminating delays and learning to continuously provision servers rapidly to shorten the time to deliver.
3. **Isolation of the build, install, configure, and customize tasks from requirements, design, and hardware setup activities:** It provides focus for leveraging provisioning automation tools.
4. **Role players performing a finite set of repeatable activities:** It enables the collection of intellectual capital necessary for beginning to automate their activities, and for planning full automation.
5. **An assembly line approach to provisioning:** It facilitates automation of piece parts of the process in an incremental approach to self-service.

Long-Term Goals

1. Achieve operational efficiencies by using a common set of processes and procedures to deliver provisioning services to the enterprise.
2. Achieve target environmental defect rate.
3. Establish and achieve service-level objectives for delivery of provisioned environments.
4. Reduce time to set up development and test environments.
5. Reduce hardware/software spending through optimization of all environments and reuse of assets.
6. Enforce enterprise provisioning standards.

Short-Term Goals

1. Reduce the defect rate for the setup of the development and test environments.
2. Improve and provide consistency in the provisioning of environments for all platforms.
3. Transfer skills and knowledge of new standard processes and procedures to the provisioning teams.

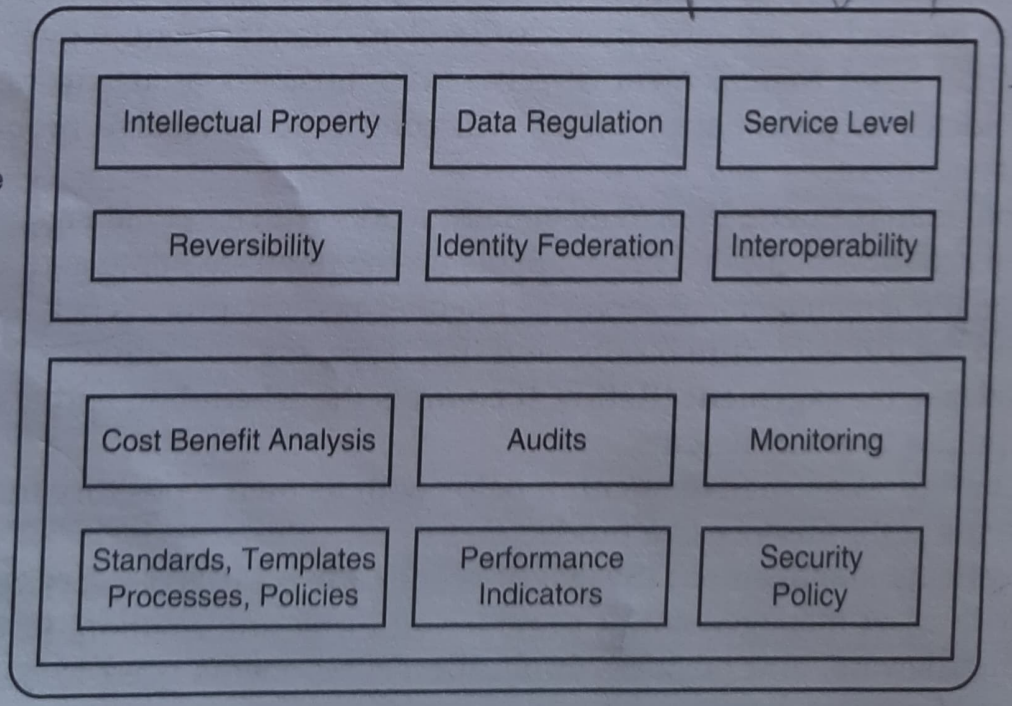
4. Gain stakeholder agreement before deployment of a provisioned product that all requirements have been met.
5. Reduce rework.
6. Improve quality of work experience for process participants.

6.5 Cloud Governance (CG)

One of the major components of any governance model is the proper definition of roles and responsibilities within an appropriate organizational structure. The domain owners within the organization own and are accountable for the business functionality within their proper business domain. These domain owners report to the head, but they also have direct reporting responsibilities within their business domain. These technical roles, along with the domain owners, strive to achieve a confluence between business and IT. One of the major aspects of cloud governance is to ensure that the lifecycle of services maximizes the value of service-oriented architecture (SOA) to the business. In order for governance to be effective, all aspects of the service lifecycle need to be properly handled.

The process transcends all phases of the service lifecycle: model, assemble, deploy and manage. Each task is numbered based on the phase it falls under. The cloud governance scenario should be broken down into realizations (see Fig. 6.1). They can be:

1. Regulation of new service creation.
2. Getting more reuse of services.
3. Enforcing standards and best practices.
4. Service management and version control.



available with respect

which are:

1. **Mean time between failures:** It is the average time for the failures occurring successively in a system.
2. **Mean time to recover:** It is the average time taken to recover from a system disaster.
3. **High availability (HA):** It is the functionality of the system that provides the agreed service levels to end-users during scheduled periods.
4. **Continuous operations:** This is the feature that gives continuous access to the end-user at any time, $24 \times 7 \times 365$.
5. **Continuous availability:** This is the characteristic to deliver the agreed service level at any time, $24 \times 7 \times 365$.
6. **Availability management:** This is the process of managing the resources such as people and technology to ensure the agreed service levels to meet the metrics and need of the organization.

implementing expanded operations

6.7.3 Cloud Chargeback Models

In consolidated environments, IT accounting service employs a cost recovery mechanism called chargeback. Chargeback is the system that is devised to put across the fee for the services provided in a cloud-based model. This allows the services in various bundles of value proposition to recover the cost for different offerings at different service levels.

In order to have the actual benefits of a chargeback model, organizations need to understand their own cost structures, and break down the components of the services and resources. The inner depth of devising an effective financial model requires functionalities such as the utility-based model to charge the various resources based on the billing costs (Fig. 6.2).

It is important to note that the chargeback models will not come with a silver lining and solve all the problems that are related to any organization's cloud commodity costing model. It comprises many proven, tested models in the industry, but each model is specifically based on

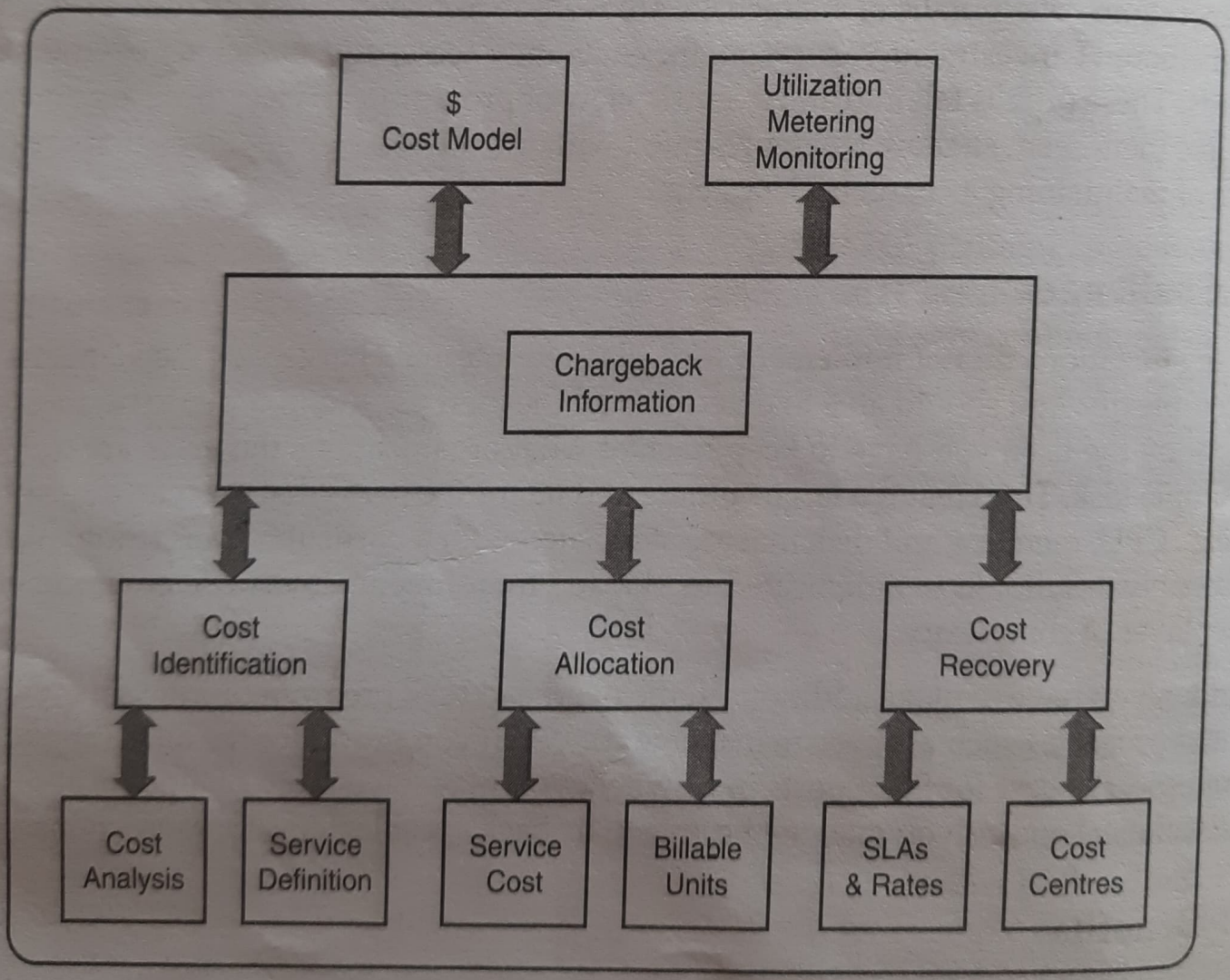


FIGURE 6.2 Chargeback model.

based
we will discuss
costing models.
that can be adopted as stand-alone or as hybrid

Subscription Model

It is the simplest model. It is actually derived by dividing the total operational cost of the organization that runs IT and the total applications hosted in the environment. The cost mechanism is very simple and recovery is easier, because of which it is more convenient to adopt. Hence it is finding its way in many organizations. This type of model requires a constant addition in the costing model by adding the upgraded cost to the model with the subscriber. This is chargeback model, but at the same time it has a drawback that it provides subsidy and unbalanced allocation of resources. In other words, some applications that are not performing well are subsidized with the high performing application and infrastructure.

Pay-Per-Use Model

This chargeback model is good in an environment where the organization has multiple lines of businesses or tenants of different size and scale. The costing model is based on the actual application consumption based on agreed SLAs. The same application can be charged differently at different service levels. If the architecture of one application is not good and consumes more resources and time, it will be charged at the upper side. It can be a complicated model as it requires managing and monitoring the service levels and resource consumption. It is good from the recovery perspective, but it is difficult to get common levels with different teams to finalize the cost and financial models.

Premium Pricing Model

In this model service and availability is guaranteed for any critical service deployment. It is devised on the line of business priorities and preferential services rendered to specific tenants. This comes at a premium pricing and supports specific policies, resource reservations, and service levels. It is also dependent on the deployment model such as a dedicated or shared model. Therefore, based on isolation and separation, a shared service-costing model can be devised. It is always preferred by those units that are involved in mission-critical applications with a high revenue generation environment. It will never exist alone in any organization. It will come in combination with other models discussed above.

'Hybrid' Model

This chargeback model is an exercise to carry the best practices of all the models and create the best costing and chargeback model for any organization. This can combine two or more chargeback models, for example, a combination of a flat fee for registering for the applications services for the first time and then charge based on the resource usage. This way we can bring transparency in the system by showcasing the flat fees as the fixed price for hosting the service, and then actual charges based on the resource utilization and usage scenarios as a variable cost.

Similarly, if we combine the subscription and as pay-per-use model, it works like the utility services provided to any home by many agencies, such as water and electricity. In total, we can say that there is no one size that fits all situations. One model can be useful for one organization and the same will not work in other organizations. It is now customary to launch service