V20PCA107 - IT INFRASTRUCTURE MANAGEMENT 2.0

UNIT - II_WEEK - 6

CAPACITY MANAGEMENT

Capacity management is the broad term describing a variety of IT monitoring, administration and planning actions that are taken to ensure that a computing infrastructure has adequate resources to handle current data processing requirements as well as the capacity to accommodate future loads. The tools used for capacity management range from spreadsheets with manually compiled performance information to the "element managers" often included with computing devices to specialized software or hardware that provides extremely detailed insights into how computing components are functioning.

Managing IT infrastructure has become more complex in recent years and companies have augmented or replaced in-house systems with resources provided by cloud services. The cloud services require the same degree of capacity management, performance management and capacity planning as on-premises gear, so more holistic capacity management and planning tools have been introduced to adequately address these hybrid environments in the management process.

Capacity Management in IT

The methodologies and processes used for IT capacity management may vary, but however it is accomplished, at minimum, it requires the ability to monitor IT resources closely enough to be able to gather and measure basic performance metrics. With that data in hand, IT managers and administrators can set baselines for operations to meet a company's processing needs. The baselines -- or benchmarks -- represent average performance over a specific period of time and can be used to detect deviations from those established levels.

Capacity Management Vs. Capacity Planning

Capacity planning is typically based on the results and analysis of the data gathered during capacity management activities.

1. By examining performance variances over time, IT management can use those performance statistics
2. to help develop models describing anticipated processing which can be used for short- and long-term planning.
3. By noting which particular resources are being stressed, current configurations can be appropriately revised and
4. IT planners can assemble purchasing plans for hardware and software that will help meet future demands.

How Capacity Management Works

Capacity management tools measure the volumes, speeds, latencies and efficiency of the movement of data as it is processed by an organization's applications. All facets of data's journey through the IT infrastructure must be monitored, so capacity management must be able to examine the operations of all the hardware and software in an environment and capture critical information about data flow. Measurement and analysis tools must be able to observe the individual performances of IT assets, as well as how these assets interact. A comprehensive capacity management process should be able to monitor and measure the following IT elements:

- Servers
- End-user devices
- Networks and related communications devices
- Storage systems and storage network devices
- Cloud services

Whether capacity management is achieved via software, hardware or manual means -- or a combination of any of those -- it relies on the interception of data movement metrics and the internal processes of individual components. Most IT hardware products ship with applications that can extract basic performance information. While the information is useful, it usually is limited and may only pertain to a few performance factors. To get more detailed statistics, an admin would typically run a software utility program designed to address specific functionalities of a components. For example, IO meter is a free, open-source utility originally developed by Intel that provides details about processing by servers, clusters of servers or individual end-user computers. One of the key metrics that IO meter provides is IOPS -- input/output operations per second -- which is a basic measure of the transfer rate of data during processing.

Emulation programs are also effective tools for capacity management. These programs mimic application programs such as database management systems (DBMSes) to determine how a system is likely to perform under similar loads in production environments. Application emulators typically include their own sets of test data to help ensure accurate and consistent results across disparate equipment.

Another approach to capacity management involves the use of hardware-based monitoring devices. Generally, these management systems focus on network performance and can provide comprehensive information on most aspects of data movement. The components of these systems vary, but a basic configuration will include control devices -- typically servers with specialized software -- and network TAPS, or network Test Access Points, devices that physically hook into particular elements of a network to capture information about data traffic as it occurs.

Components of Capacity Management

Capacity management could have a fairly narrow scope, providing high-level information on a variety of infrastructure components or, perhaps, providing detail metrics related to one segment of the computing environment. The trend, however, is to gather as much information as possible and then to attempt to correlate those measurements into an application-centric picture that focuses on the performance and requirements of mission-critical applications across the environment, rather than how individual components are performing. Still, to achieve that application-centric view of capacity management, virtually all elements of the IT infrastructure must be monitored and the definition of capacity must be broad enough to consider the impact an application will have on processing power, memory, storage capacity and speed for all physical and software components comprising an infrastructure.

Performance -- or throughput -- is a key metric in capacity management as it may point to processing bottlenecks that affect overall application processing performance. The central processor unit (CPU) in servers and other connected devices, such as routers, storage and controllers, should be monitored to ensure that their processing capabilities are not frequently "pinning" at or near 100%. An overtaxed processor would be a candidate for upgrading.

Memory is also a factor in capacity management. Servers and other devices use their installed memory to run applications and process data -- if too little memory is installed, processing will slow down. It's relatively easy to determine if a server has adequate memory resources, but it's also important to monitor other devices in the environment to ensure that insufficient memory doesn't turn them into processing bottlenecks.

Physical space is what is most associated with capacity management, with the focus generally on storage space for applications and data. Storage systems that are near capacity will have longer response times, as it takes longer to locate specific data when drives -- hard disk or solid-state -- are full or nearly full. As with processor and memory measurements, it's important to monitor space usage in devices other than servers and end-user PCs that may have installed storage that's used for caching data.

Capacity Management in Networking

Managing the capacity of IT networks can be a complex process given the number of different networking elements that can be found in an enterprise environment.

The number and type of networks being monitored is likely to vary as well. In addition to the wired and wireless Ethernet-based network infrastructure that connects servers to storage, end-user devices, networking gear, etc., comprehensive network capacity management must also consider dedicated storage networks based on Fibre Channel technologies; the FC networks are likely to be physically isolated from other data networks and will require different tools for monitoring and management.

External networking should also be monitored. Again, different tools will be required to track traffic and performance for network connections to remote offices and users, the internet and to cloud services.

The networking devices that should be monitored include network interface cards (NICs), network switches, network routers, storage network interfaces (e.g., host bus adapters), storage network switches and optical network devices.

Although capacity management for networks doesn't directly address security, it can be a good method of keeping track of network access, which can help inform security procedures.

Benefits of Capacity Management

Capacity management provides many benefits to an IT organization and is a factor in overall management of a computing infrastructure. In addition to ensuring that systems are performing at adequate levels to achieve a company's goals, capacity management can often realize cost savings by avoiding over-provisioning of hardware and software resources. It can also help save money and time by identifying extraneous activities like backing up unused data or maintaining idle servers. Good capacity management can also result in more-effective purchasing to accommodate future growth by being able to more accurately anticipate needs and, thus, make purchases when prices may be lower. By constantly monitoring equipment and processing, problems that might have hindered production may be avoided, such as bottlenecks or imminent equipment failures.

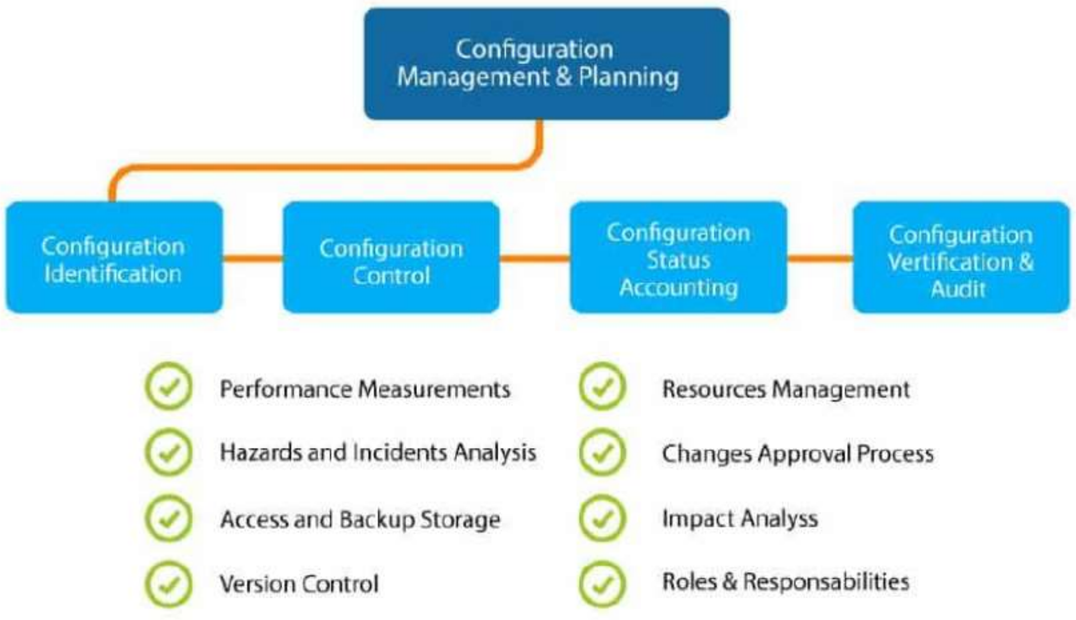
CONFIGURATION MANAGEMENT

Configuration management is an increasingly important foundation for a successful tech platform. Good leaders in the tech space will want to know what it takes to implement it. If that's you, you're in luck! In this post, we'll discuss a few key things:

- What configuration management is and where it originated from.
- The benefits of configuration management.
- How configuration management fits with concepts like DevOps and agile.
- How you can get started with configuration management.

Sound good? Let's get going.

Configuration Management Here's my definition of configuration management: it's the discipline of ensuring that all software and hardware assets which a company owns are known and always tracked—any future changes to these assets are known and tracked. You can think of configuration management like an always up-to-date inventory for your technology assets, a single source of truth.


Configuration Management & Planning
Configuration Identification
Configuration Control
Configuration Status Accounting
Configuration Vertification & Audit
Performance Measurements
Hazards and Incidents Analysis
Access and Backup Storage
Version Control
Resources Management
Changes Approval Process
Impact Analyss
Roles & Responsabilities

Configuration Management & Planning

With that defined, let's talk about how it works in practice. Configuration management usually spans a few areas. It often relates to different ideas, like creating "software pipelines" to build and test our software artifacts. Or it might relate to writing "infrastructure-as-code" to capture in code the current state of our infrastructure. And it could mean incorporating configuration management tools such as Chef, Puppet, and Ansible to store the current state of our servers.

Configuration Management Originate

When I first started learning about configuration management, I found the concept super confusing. However, it turns out that there are reasons for the confusion. But to understand why, we need to look at some history. The idea of configuration management comes from other institutions, such as the military. We took those ideas and retrofitted them into a software context.

How the Software Has Changed Over Time

Configuration management was traditionally a purely manual task, completed by a systems administrator. The role was a lot of manual work involving carefully documenting the state of the system. But the industry has changed completely. These changes came from the popularity of DevOps, increases in cloud computing, and new automation tooling. Now that we've set the scene, we can dive into the details of configuration management. So let's get to it!

What the World Looks Like With Configuration Management Before we explore different tools for configuration management, we need to know what end results we'll receive for our efforts.

What are the outcomes of well-implemented configuration management?

Let's cover the benefits.

Benefit 1: Disaster Recovery If the worst does happen, configuration management ensures that our assets are easily recoverable. The same applies to rollbacks. Configuration management makes it so that when we've put out bad code, we can go back to the state of our software before the change.

Benefit 2: Uptime and Site Reliability The term "site reliability" refers to how often your service is up. I've worked at companies where each second of downtime would cost thousands—often tens or even hundreds of thousands. Eek! A frequent cause of downtime is bad deployments, which can be caused by differences in running production servers to test servers. With our configuration managed properly, our test environments can mimic production, so there's less chance of a nasty surprise.

Benefit 3: Easier Scaling Provisioning is the act of adding more resources (usually servers) to our running application. Configuration management ensures that we know what a good state of our service is. That way, when we want to increase the number of servers that we run, it's simply a case of clicking a button or running a script. The goal is really to make provisioning a non-event. These are just some of the benefits of configuration management. But there are some other ones, too. You'll experience faster onboarding of new team members, easier collaboration between teams, and extended software lifecycle of products/assets, among other benefits.

The World Without Configuration Management

Sometimes it's easier to grasp a concept by understanding its antithesis. What does trouble look like for configuration management, and what are we trying to avoid? Let's look.

A developer implementing a feature will commonly install a few bits of software and deploy code. If things are sloppy, this developer probably makes the team and manager aware of the intention to come back later to clean it all up—that it's simply a demonstration and will be rewritten soon. But then the deadline starts pressing, and the task of going back through and rewriting the installation steps as a script gets pushed lower and lower in priority. Before we know it, several years have passed, and a new developer gets put on the project. That developer is now left to pick up the pieces, trying to understand what happened. It's quite likely they aren't even going to touch the configuration of the server. Who knows what it would do! The above situation is precisely what configuration management helps you avoid. We don't want to be left in the dark as a result of developers setting up software without proper documentation/traceability. Rather, we want to know the answers to questions like

- What services are we running?
- What state are those services in?
- How did they get to their current state?
- What was the purpose for the changes?

Configuration management can tell us these answers. That hopefully paints a clearer picture of the problems that configuration management is trying to solve.

Configuration Management Fits in With DevOps, Continuous Delivery, and More… Hopefully by now you're starting to get the hang of what configuration management is and what it aims to do. Before we go on to discuss tooling, I'd like to take a moment to address how configuration management fits in with other software development concepts like agile, DevOps, continuous integration, continuous delivery, and Docker so that you can understand how these concepts fit in with the ideas of configuration management.

Is Configuration Management Compatible with Agile?

Yes. Agile software, by definition, reflects the desire to make changes to our software faster so that we can respond to market demands. Configuration management helps us to safely manage our changes and keep velocity high.

How Does Configuration Management Fit with DevOps?

DevOps is the extension of agile practices across both the development and operations departments. In fact, DevOps seeks to unify the goals of both departments. At some companies, the development department seeks change while the operations department seeks stability. But companies that embrace DevOps want both stability of their deployed assets and frequency of change. However, achieving this outcome requires cultural change. Like agile, configuration management gives teams the confidence to move quickly with their changes. Under agile practices, the company gives configuration management responsibilities to the development teams, empowering them to provision, configure, and manage their own infrastructure. You build it, you run it.

Where Do Pipelines Fit into Configuration Management"?

Software pipelines are the steps (or "value stream," which we can create with tools like Plutora) that we usually automate, taking code from commit to production. Pipelines usually involve steps such as linting code, unit testing code, integration testing code, and creating artifacts. A software pipeline therefore is a form of configuration management. When we build software with tools like Docker, we codify our build instructions into our Dockerfile. This allows us to better understand the dependencies of our artifacts.

Is Infrastructure-as-Code Configuration Management?

Infrastructure-as-code (or IaC for short) is the practice of ensuring all provisioned infrastructure is done so through code. The purpose of IaC is to have a written record of which services exist, where they are located, and under what circumstance. Configuration management might choose to leverage aspects of IaC to achieve the full understanding of all the technology assets a company owns.

Is Continuous Integration/Delivery Configuration Management?

Continuous delivery is the process of ensuring that software is always in a releasable state. You can achieve this through heavy automation and testing. Continuous integration is the process of bringing separate software artifacts together into a single location on a frequent basis, for the purposes of verifying that the code integrates properly. Continuous integration tools, which are typically servers that run automation-testing suites, act as a form of configuration management by providing visibility into the steps required to set up and configure a given software artifact. That should clear up some of your lingering questions about how configuration management fits with some practices or ideas that you might be using or are familiar with. Any discussion of configuration management would be incomplete, however, without a discussion about tooling. So, let's take a peek at the different tools we have at our disposal for implementing configuration management.

The Importance of Declarative Style in Configuration Management Tools

Next up, we're going to discuss configuration management tools. But before we get to that, I need to quickly discuss a concept to consider when comparing tools. And the concept is declarative style. You'll hear about this terminology a lot if you go out and start looking into different configuration management tools. So, it makes sense to have a firm grasp of what declarative style is, why it's important, and why so many people are talking about it. So, what do we mean by declarative style? And why is declarative style so important for configuration management?

What Do We Mean by Declarative Style?

When it comes to software, having a declarative style means telling your software the end result you want and then letting the software do the work in figuring out the way to get there. The opposite of the declarative style would be a procedural style, where instead of giving an end state, you give instructions on how to get there. The problem with instructions is that they're dependent on the starting state. You can think of it like this: declarative versus procedural is the difference between giving a friend your home address and giving them step-by-step instructions to get to your house from where they are. The problem with giving step-by-step instructions is that it assumes you know where the friend is starting, and it doesn't allow for things to go wrong. It's hard to replay steps when you're in a bad state (i.e., lost!).

Why Is Declarative Style Important for Configuration Management?

By now, you're probably thinking that declarative style sounds interesting. But why is it important? Declarative style is important because configuration management is all about knowing the current state of your applications. So when we use configuration management tools, it's desirable to use a declarative style and specify the end result that we want, not the steps to get there. This means we always know what end state we're trying to achieve and how that's changed over time. That's instead of trying to work out when instructions were run and dealing with the complexities that may arise if certain instructions have failed.

What Are Configuration Management Tools?

There are many different tools for configuration management. In fact, it can get confusing, as there are tools that support configuration management without explicitly being configuration management tools. For instance, Docker neatly packages up steps needed to set up and run an application (in a Dockerfile). However, people don't often consider Docker a configuration management tool. To make things clearer, let's divide up the common tools that might fall under or relate to configuration management:

Configuration Management Tools

These are the tools you see typically associated with configuration management. Tools like Chef, Ansible, and Puppet provide ways to codify steps that we require in order to bring an asset in line with a current definition of how that asset should look. For instance, you might create an Ansible playbook that ensures that all of our X servers have Y installed within them.

Infrastructure-as-Code Tools

Often also called provisioning tools, IaC tools include CloudFormation and Terraform. If our configuration management tools include the setup we need on our assets, our provisioning tools are how we get those assets. It's this blurred line that explains why we need to bring these tools into our discussion of configuration management. And many consider it an anti-pattern to use configuration management tools for provisioning.

Pipeline Tools

We talked briefly about software delivery pipelines but implementing them requires tooling. Popular technologies include Jenkins, CircleCI, and GitLab CI. By using tools to codify our build process, we make it easy for other developers to understand how our artifacts are modified and created, which is a form of configuration management.

Source Control Tools

Source control tools include GitHub, SVN, GitLab, and Bitbucket. While we need to codify our automation in scripts, if we don't appropriately track the history of our changes, then we aren't really achieving configuration management. We're now nearing the end of our introduction to configuration management. We've covered what configuration management is, we know the benefits, and we're now up to date on the latest tools. However, all of this information can be a little overwhelming if you're asking the simple question of "Where should I start?" Let's break it all down so that you can start your journey into configuration management.

AVAILABILITY MANAGEMENT

The Goal of Availability Management is to ensure that level of service availability is delivered in all services is matched to or exceeds the current and future agreed needs of the business in a cost-effective manner. Availability is one of the most critical parts of the warranty of a service. If a service does not deliver the levels of availability required, then the business will not experience the value that has been promised. Without availability the utility of the service cannot be accessed. Availability management process activity extends across the service lifecycle.

The Objectives of Availability Management Process are to:

• Produce and maintain an appropriate and up to date availability Plan, that reflects the current and future needs of the business.

• Provide advice and guidance to all other areas on availability related issues.

• Assist with diagnosis and resolution of availability related incidents and problems.

• Asses the impact of all changes on the Availability Plan, and the performance, and capacity of services and resources.

• Ensure that proactive measures are implemented to improve the availability of services wherever it is cost justifiable. In short, Availability management should always ensure that the agreed level of availability is provided. The measurement and monitoring of IT availability is a key activity to ensure availability levels are being met.

Availability Management – Basic Concepts

Let's have a look at few key concepts of Availability Management.

Availability: In simplest terms, the ability of a service, component or CI to perform its agreed function when required. This term answers the question, is it available to use when needed? **Reliability:** A measure of how long a service, component or CI can perform its agreed function without interruption. When we understood the availability, the questions come, how long the service will be available? This can be answered by reliability.

Maintainability: A measure of how quickly and effectively a service, component or CI can be restored to normal after a failure. The emphasis is on how soon or how quickly the services can be back to BAU.

Serviceability: The ability of a third-party supplier to meet the terms of their contract. Often this contract will include agreed levels of availability, reliability and / or maintainability for a supporting service or component. Serviceability and Maintainability are often confusing terms however the key differentiating term here is involvement of third-party suppliers to restore the service or component.

Understanding Basic Concepts of Availability Management

Let's understand these four terms Availability, Reliability, Maintainability, Serviceability Take an example of typical incident Lifecycle.

• The moment incident is detected, and it is restored, delta of that time is called MTTR or Mean Time to Repair and can be termed as the Serviceability IF a third-party supplier was involved.

• After restoration to the occurrence of next incident, the time when service was available to use is known as MTBF or Mean Time Between Failures and can be termed as Availability.

• Maintainability can be explained by time taken to repair, shorter the time, greater the maintainability.

• And finally, time between the two incidents is known as MTBSI or Mean Time Between System Incidents and can be termed as Reliability. Components with a high reliability factor mostly remain in high demand like some companies provide MTBSI rating along with their networking components such as routers.

Availability Management – Key Role

The Availability Management process includes two types of activities:

Reactive Activities

These involve monitoring, measuring, analysis and management of all events, incidents and problems involving unavailability. These activities are principally performed as part of the operational roles.

Proactive Activities

These involve the proactive planning, design and improvement of availability. These activities are principally performed as part of the design and planning roles. The key role of Availability Manager who is expected to involve in the activities like:

Responsible for Availability Management process

- Participate in IT infrastructure design
- Monitor actual IT availability achieved
- Create, maintain & review AMIS and Availability Plan
- Availability testing schedule
- Testing after major business change
- Assess impact of changes on Availability Plan
- Attending CAB meetings
- Ensure cost justified levels of IT availability Assessment and management of risk.

Availability Management - Techniques

Availability Management's main objective is to ensure the agreed availability being provided to the customer or the end users. But due to many reasons this assurance may get affected. There are some techniques that's being used to understand the reason for the disruption of availability. Let us begin with: Component Failure Impact Analysis (CFIA) can be used to predict and evaluate the impact on IT service arising from component failures within the technology. The output from a CFIA can be used to identify where additional resilience should be considered to prevent or minimize the impact of component failure to the business operation and users. Single Point of Failure analysis A Single Point of Failure (SPoF) is any component within the IT infrastructure that has no backup or fail-over capability, and has the potential to cause disruption to the business, customers or users when it fails.

It is important that no unrecognized SPoFs exist within the IT infrastructure design or the actual technology, and that they are avoided wherever possible. Fault Tree Analysis Fault Tree Analysis (FTA) is a technique that can be used to determine the chain of events that causes a disruption to IT services. FTA, in conjunction with calculation methods, can offer detailed models of availability. This can be used to assess the availability improvement that can be achieved by individual technology component design options. Using FTA: • Information can be provided that can be used for availability calculations • Operations can be performed on the resulting fault tree; these operations correspond with design options • The desired level of detail in the analysis can be chosen.

Availability Management - Challenges and Risks

Availability management faces many challenges, but the main challenge is to actually meet and manage the expectations of the customers, the business and senior management. These expectations are frequently that services will always be available not just during their agreed service hours, but that all services will be available on a 24-hour, 365-day basis. When they are not, it is assumed that they will be recovered within minutes. This is only the case when the appropriate level of investment and design has been applied to the service, and this should only be made where the business impact justifies that level of investment. However, the message needs to be publicized to all customers and areas of the business, so that when services do fail they have the right level of expectation on their recovery. It also means that availability management must have access to the right level of quality information on the current business need for IT services and its plans for the future. This is another challenge faced by many availability management processes. Another challenge facing availability management is the integration of all of the availability data into an integrated set of information (AMIS) that can be analysed in a consistent manner to provide details on the availability of all services and components. This is particularly challenging when the information from the different technologies is often provided by different tools in differing formats. Yet another challenge facing availability management is convincing the business and senior management of the investment needed in proactive availability measures. Investment is always recognized once failures have occurred, but by then it is really too late. Persuading businesses and customers to invest in resilience to avoid the possibility of failures that may happen is a difficult challenge. Availability management should work closely with ITSCM, information security management and capacity management in producing the justifications necessary to secure the appropriate investment.

Some of the major risks associated with availability management include:

• A lack of commitment from the business to the availability management process

• A lack of commitment from the business and a lack of appropriate information on future and strategies

• A lack of senior management commitment or a lack of resources and/or budget to the availability management process.

• Labour-intensive reporting processes

• The processes focus too much on the technology and not enough on the services and the needs of the business

• The AMIS is maintained in isolation and is not shared or consistent with other process areas, especially ITSCM, information security management and capacity management. This investment is particularly important when considering the necessary service and component backup and recovery tools, technology and processes to meet the agreed needs.

RELEASE MANAGEMENT:

Release Management is all about enabling an organization's systems and services to change to support evolving business needs. It is the process of coordinating the movement of project into production environments where they can be consumed by end-users. The primary goal of release management is to ensure that the integrity of the live environment is protected and that the correct components are released. In some organizations, release management is concerned only with the technical deployment of IT products and features, while other organizations take a broader perspective of release management also including things like adoption and business process changes related to a release.

Release Management in ITIL

Release and Deployment Management is one of the main processes under the Service Transition section of the IT Infrastructure Library (ITIL®) framework. This process is often referred to in short form as simply “release management”. ITIL defines release and deployment management as the process of managing planning and scheduling the rollout of IT services, updates and releases to the production environment. Release in this context refers to the development of a newer version of a service or component and deployment means the process of integrating it into the live production environment. Release management plays an important role of bridging the gap between project activities and the things that project teams produce and the ongoing operations and users that will consume these things. It is very common for organizations to have multiple projects underway at the same time and release management provides a structured approach for bringing changes together, testing to make sure they work correctly and then safely introducing them into the live environments that business operations rely on. Release management also ensures that any applicable knowledge and resources are transferred from the teams developing the new features or components to the operations team that will be responsible for supporting them.

The Release Management Process

It is important that each project team wishing to introduce changes to the production environment are aligned with each other and are aware of each other’s changes and resource usages. They must follow the same process, policies and guidelines for planning, building, testing and deploying a release. ITIL breaks release management down into six sub-processes that enable release management to be performed effectively, efficiently and safely to facilitate the flow of changes into the operations environment.

- **Release management support**: provides guidelines and support for the deployment of releases including the roles that are involved in other parts of the release and deployment management process

- **Release planning:** defines the scope and content of releases according to release management policies, assigns authorized changes into release packages and defines a schedule for building, testing and deploying the release

- **Release build:** deals with the actual development of all required release components including the issuance of all necessary work orders and purchase orders for components sourced from vendors and ensuring that all release components are ready for validation and testing

- **Release deployment:** manages the deployment of release components into the live production environment and the transition of documentation and training to end-users and operating staff.

- **Early-life (post-release) support:** the initial period after the deployment of a new release when the release and deployment management team work with the incident management team to resolve operational issues and remove errors and deficiencies caused by the release.

- **Release closure:** formally closing release activities, verifying all documents and records are properly updated and reporting release outcomes and feedback to project teams.

The Role of a Release Manager

The release manager plays a combined coordination and governance/oversight role – tasked with ensuring that the release is completed effectively and safely. Release managers are typically IT professionals with specialized skills and experience using standards, processes and tools to coordinate release activities. In IT contexts, release managers work with business leaders, IT project teams and operations staff to ensure a well-orchestrated release of technical features into the IT environment. In product management contexts, release managers work with business development, marketing, R&D and other teams to coordinate across the company in support of a planned product release.

Large releases may involve multiple staff members working as a release team. If this is the case, the release manager provides an over-arching management and leadership function, coordinating both the release team and the release itself. The release manager is responsible for governance and quality oversight of the release, determining what level of risk and complexity the release represents and ensuring that the right level of due-diligence is applied to ensure release objectives are achieved without compromising the ongoing operations of the company.

Why is Release Management Needed?

The primary objective of release management is to plan, schedule and control the deployment of IT services and updates into the live environments. Companies evolve and as they do, their needs change, so the IT environment needs to change too. Release management provides a means for making changes effectively and safely. It does this by ensuring only sufficiently tested services and components can be released into the live environment that the business uses. Some other benefits of release management include:

- Faster delivery of changes and new features to users
- Reduced risk of un-authorized releases breaking features that people are using
- Predictable schedule of deployments at times that minimize business impact
- Ensuring new or changed services can meet agreed service requirement
- Providing proper knowledge transfer to users and support staff

Release and Deployment Management Approaches

ITIL v3 defines six approaches for release and deployment management. Most companies utilize some variant of these approaches though they may refer to them by different names. It is also common for different approaches to be used for different types and sizes of projects.

• Big Bang Approach: new or changed services is deployed to all users at the same time.

• Phased Approach: services are initially deployed to a part of the user base and if no issue observed, the deployment is repeated to other user groups via a scheduled rollout plan

• Push Approach: the service component is deployed from a central location and pushed out to the target audience/locations at a pre-defined time

• Automated Approach: changes are deployed into the production environment using automated workflows and distribution mechanisms.

• Manual Approach: frelies on manual activities to distribute a release (often used when the release has system dependencies that require manual checking before or after deployment)

Release and Deployment Management is tasked with enabling an organization's systems and services to change to support evolving business needs. There may be multiple project teams and vendors working on developing individual changes and it is releasing management's job to ensure that all the pieces of the puzzle come together and are tested appropriately prior to introducing them into the live environment that business users rely on. Release management processes coordinated by a skilled and experienced release manager who is equipped with the right set of tools to do their job effectively can all help a company orchestrate safe and effective changes to their IT environment.