Release Management

What is Release Management?

- Release Management is the process responsible for planning, scheduling, and controlling the build, in addition to testing and deploying Releases.

- Release management usually begins in the development cycle with requests for changes or new features.

- If the request is approved, the new release is planned and designed.

- The new design enters the testing or quality assurance phase, in which the release is built, reviewed, tested until it is ultimately accepted as a release candidate.

- The release then enters the deployment phase, where it is implemented and made available.

- Once deployed, the release enters a support phase, where bug reports and other issues are collected; this leads to new requests for changes, and the cycle starts all over again.

- **Objectives:**

- Increase the number of successful Releases, including reducing the number of Releases with unexpected outcomes.

- Decrease the number of incidents caused by Releases.

- Create a single documented process for managing all Releases.

- Maintain a single repository for recording all Releases through the lifecycle.

- System release managers are responsible for:

  1. Deciding when the system can be released to customers

  2. Managing the process of creating the release and distribution media

  3. Documenting the release

System releases

- Not just a set of executable programs
- May also include
  - Configuration files defining how the release is configured for a particular installation
  - Data files needed for system operation
  - An installation program or shell script to install the system on target hardware
  - Electronic and paper documentation
  - Packaging and associated publicity
- Systems are normally released on CD-ROM or as downloadable installation files from the web

Release problems

- Customer may not want a new release of the system.
  - They may be happy with their current system as the new version may provide unwanted functionality (may not be user-friendly).
- All files required for a release should be re-created when a new release is installed

1. Release decision making

- Preparing and distributing a system release is an expensive process.

- Factors such as the technical quality of the system, competition, marketing requirements and customer change requests should all influence the decision of when to issue a new system release.

2. Release creation

- Release creation involves collecting all files and documentation required to create a system release.
- Configuration descriptions have to be written for different hardware and installation.
- Instructions prepared for customers who need to configure their own systems.
- Release directory is handed over for distribution.

3.Release documentation

- The specific release must be documented to record exactly what files were used to create it.
- To document a release
  - Have to record the versions of the source code components that were used to create the executable code
  - Keep copies of source and executable code and all data and configuration files
  - Record versions of the OS ,libraries, compilers and other tools

Monitoring – Continuous Monitoring

- In a DevOps environment, failures must be found and fixed in real time. How do you do that?

- A big part is monitoring/continuous monitoring.

- Goals of continuous monitoring are:

  - to quickly determine when a service is unavailable,
  - understand the underlying causes
  - most importantly, learn/understand them properly to anticipate problems before they occur.

- Some monitoring experts advocate that the definition of a service must include monitoring—they see it as integral to service delivery.

- Like testing, monitoring starts in development.

- Two kinds of monitoring are required for DevOps:

  1. server monitoring
  2. application performance monitoring.

- **Server Monitoring**

- Server Monitoring is a process to monitor server's system resources like CPU Usage, Memory Consumption, I/O, Network, Disk Usage, Process etc.

- In short, server monitoring helps understanding system's resource usage.

- **Application Performance Monitoring**

- Application performance monitoring (APM) is an area of information technology that focuses on making sure software application programs perform as expected.

- The goal of performance monitoring is to provide end users with a quality end-user experience.