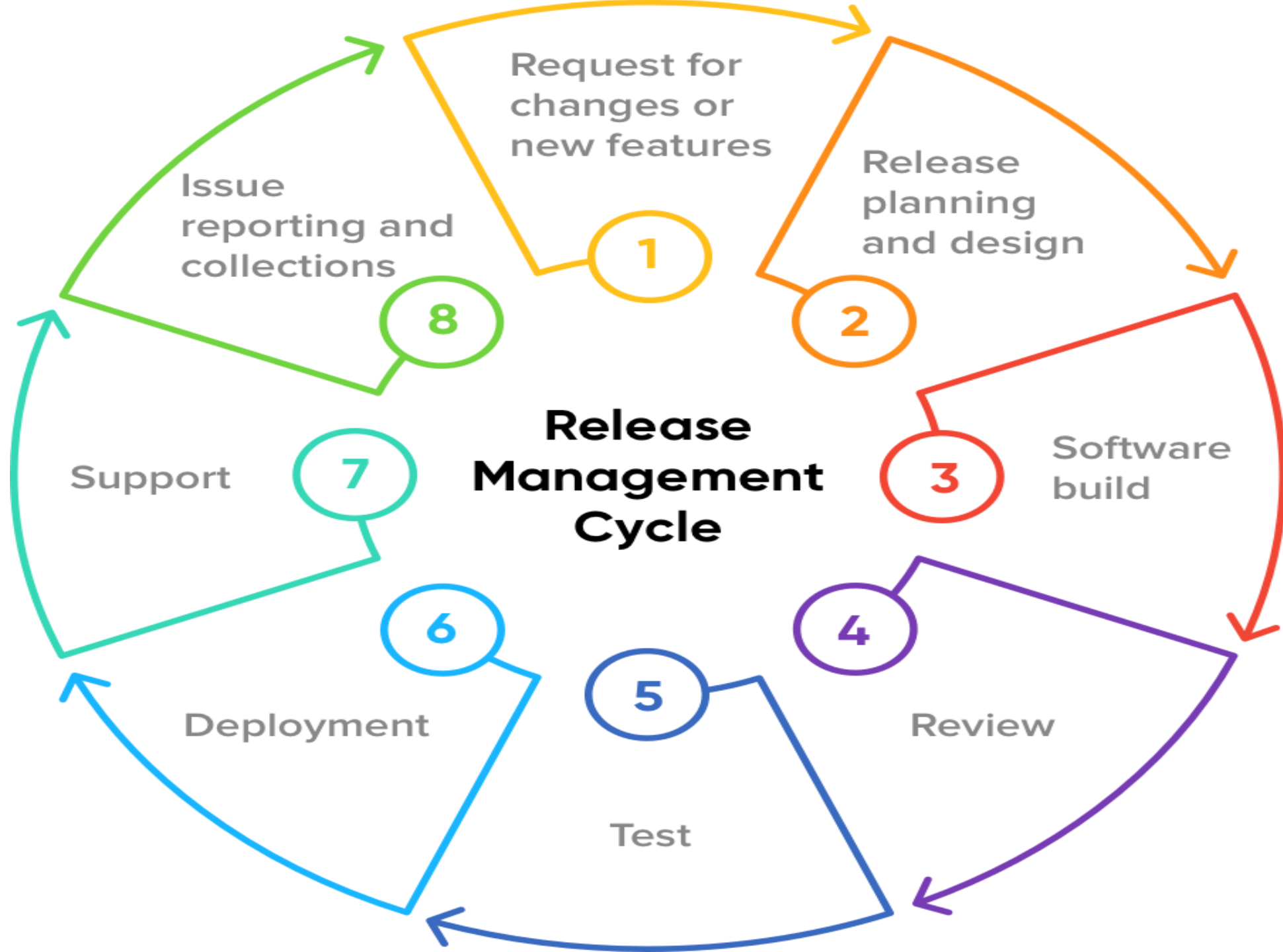


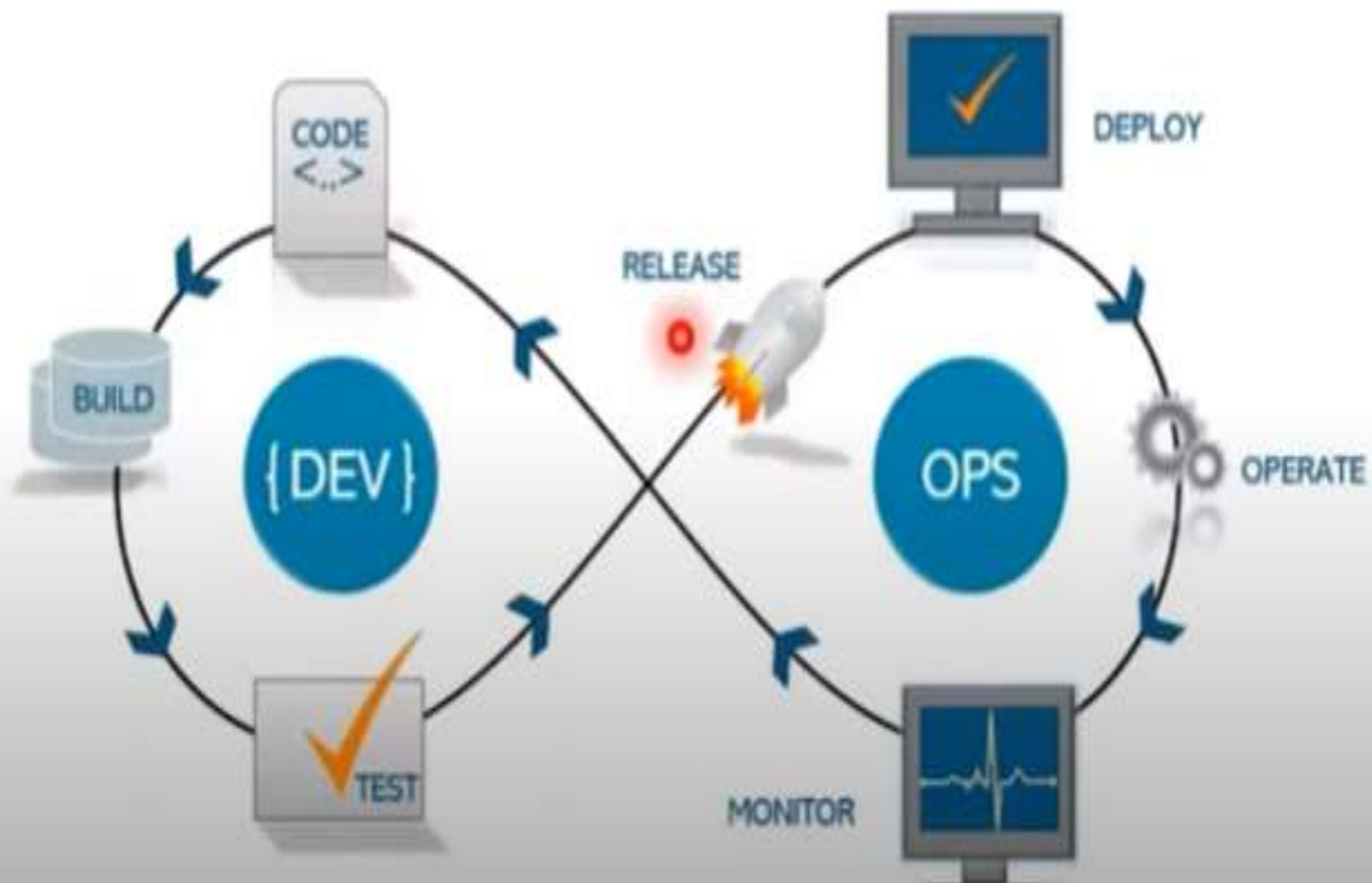
Release Management

What is Release Management?

“Release management is the process of managing, planning, scheduling and controlling a software build through different stages and environments; including testing and deploying software releases”

Source: Wikipedia







- ✓ Release management in DevOps is about planning, scheduling and controlling the software development and delivery process.
- ✓ IT operations and Developers collaborate from the beginning of the process to the end in DevOps for shorter and fewer feedback loops and faster releases.

What is DevOps Release Management?

What is Release Management?

Release Management is the process that handles software deployments and change initiatives. Across an organization, it schedules the relevant tasks (internal and external), assigns the physical and human resources needed to carry them out, and oversees the execution. It starts with planning what will be contained within a release, managing the software build through different stages and environments, testing stability and finally, deployment.

It is a difficult balancing act between:

- Delivering value and novel features to customers
- Prioritizing value streams according to business needs
- Improving the speed and quality of release deployments
- Not compromising product stability.

No matter the software development approach, the Release Manager has the skill set, initiative and determination to make this happen.

Release Management Process Components



Release Management Process Components

- **Release Value Stream:** The release processes that add or create value across the release pipeline.

- To understand value stream mapping, we need to first understand what a “value stream” is. Simply put, **a value stream is a series of steps that occur to provide the product or service that their customers want or need.** In order to provide the product or service that the customers desire, every company has a set of steps that are required.

- **Release Pipeline:** A specific release process from feature planning to delivery.

- A release pipeline is **a conceptual process by which we take committed code into production.** As such, a release pipeline can be as ephemeral or as real as we want to make it. The fundamental release pipeline from code change to production software. A means of triggering the pipeline to run.

- **Release Policy:** The definition of release types, standards, governance requirements(set of rule) for an organization.

- The Release Policy represents **a set of rules for deploying releases into the live operational environment,** defining different approaches for releases depending on their urgency and impact.

- **Release Template:** A single, repeatable workflow process for release pipeline that includes human and automated activities and follows an organization's release policies
- **Release Plan:** An instance of a release template developed for a specific release.

RELEASE PLAN TEMPLATE



- **Deployment Plan:** Activities to deploy a release to the production environment.



stakeholdermap.com

Project Management, project planning, templates and advice

DEPLOYMENT PLAN TEMPLATE

| Task | Start date & time | Finish date & time | Owner | Status | Comments/progress |
|--|--------------------|--------------------|--------|-------------|---|
| Confirm completion of user training at board meeting. | [dd/mm/yyyy hh:mm] | [dd/mm/yyyy hh:mm] | [name] | Completed | Trainer will 'walk the floor' on go live day to support users. |
| Stand up the production environment | [dd/mm/yyyy hh:mm] | [dd/mm/yyyy hh:mm] | [name] | In progress | |
| Run deployment wizard | [dd/mm/yyyy hh:mm] | [dd/mm/yyyy hh:mm] | [name] | Open | |
| Check error log & resolve any issues manually if needed. | [dd/mm/yyyy hh:mm] | [dd/mm/yyyy hh:mm] | [name] | Open | |
| Complete manual configuration items. | [dd/mm/yyyy hh:mm] | [dd/mm/yyyy hh:mm] | [name] | Open | See production only configuration section in technical design document. |
| Review production. | [dd/mm/yyyy hh:mm] | [dd/mm/yyyy hh:mm] | [name] | Open | Quality check by super users. |

- **Release unit:** The set of artifacts(product backlog, burndown charts) released together to implement a specific feature.
 - A Release Unit is a **set of new, changed and/or unchanged Configuration Items**, which are tested and introduced into the live environment together to implement one or several approved Changes.
- **Release package:** A combination of one or more release units deployed together as a single release due to interdependencies, scheduling, or business priorities.
 - A release package contains **information about all of the configuration items** (for example, documentation, test plans, services, and so on) associated with a change and links to the release that is deploying the change. Typically, a Change Manager might have already linked to a release, or created a request for release.

- **Major Releases:** Infrequent release packages that often include many release units that have a high or critical business impact.

- **Minor Releases:** More frequent release packages with fewer release units that do not include mission critical components.

- A major release consists of major new features and/or large architectural changes. ... Some existing features may be deprecated. A minor release includes some new features and functions. The release may include limited architectural changes that enhance sustainability and performance.

Release Pipeline

A particular release process from feature planning to delivery

Release Value Stream

The release processes that add or create value across the release pipeline



Release Management Process Components

Release Policy

Release policy consists of the definition of release types, standards and governance requirements for an organization

Release Template

Release template is a single or repeatable workflow process for release pipeline which includes human and automated activities.



Release Management Process Components

Release Plan

A structured plan of an instance of a release template developed for a specific release

Deployment Plan

Deployment Plan consists of activities to deploy a release to the production environment.



Release Management Process Components

Release Unit

The set of artifacts released together to implement a specific feature

Release Package

Release package is a combination of release units deployed together as a single release due to interdependencies, scheduling, or business priorities.



Release Management Process Components

Major Release

Infrequent release packages that include many release units that have a high or critical business impact.

Minor Releases

More frequent release packages with fewer release units that do not include critical components.



Release Management Process Components

Strategies of Release Management



Release Windows (slow cadence)

A release window is a period of time in which one or more teams release into production.

Release Train



Strategies of Release Management

There are four general [release scheduling](#) strategies that potentially support DevOps. These strategies, from least effective to most effective, are:

1. **Release windows (slow cadence)**. A release window is a period of time during which one or more teams may release into production.
 - ✓ A release slot is subset within that release window (and may be the entire window) during which a team may deploy their solution into production. For example, your organization may have a policy that production releases occur between 1am and 5am on Saturday evenings (the release window) and that up to four releases may occur during that window (the release slots).
 - ✓ In lean terms, a release slot is effectively a Kanban card that allows a team to deploy. Release windows tend to align with periods where system usage is lower.
 - ✓ With a slow cadence(tempo) approach to this strategy the release windows occur far apart, as seldom as once a week or even longer. The advantages of this approach are that it provides a consistent release cadence to business stakeholders and it provides consistent release date targets for delivery teams.
 - ✓ The primary disadvantage with slow cadence release windows is that they become bottlenecks for release management processes that need to support multiple teams. There are only so many release slots available during each window and this number can be easily exceeded, forcing teams to aim for a future release window. This problem becomes exacerbated when teams start to move to a continuous delivery strategy.

- ✓ **Release train.** The idea with a release train is that every team involved with that “train” has the same release cadence — [for example this train releases once a quarter, or once a month](#), or even once a week. This strategy is commonly used in large programs, or teams of teams, where the individual teams are each working on part of a larger whole.
- ✓ Having the common drumbeat of a release train provides a consistent release schedule for stakeholders and serves as a rallying point for development teams. The train metaphor works quite well in practice.
- ✓ [If your team misses the release date, if you miss the train, then the train goes on without you and you need to wait for space on the next on.](#)
- ✓ Dependencies are also respected. For example, if several components need to ship together they must all go on the same train (similar to a family taking a trip together).
- ✓ [The primary disadvantage is that development teams are constrained to a common release schedule, making it difficult to support lean or continuous delivery strategies.](#)
- ✓ A potential disadvantage is that release trains may also suffer from the bottleneck problems of slow cadence release windows.

Release Windows (quick cadence)

The quick cadence release are less likely to suffer from the bottleneck challenges associated with slow cadence release windows and release trains.

Continuous Release Availability

With this approach delivery teams are allowed to release their solutions into production whenever they need to.



Strategies of Release Management

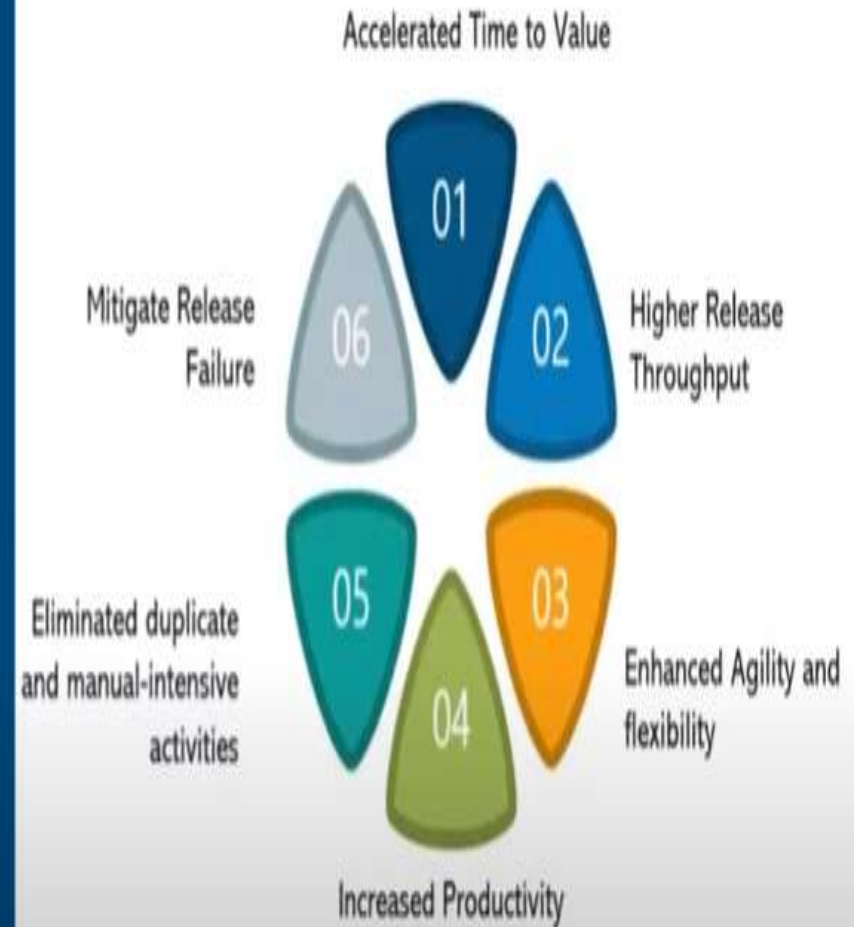
Release windows (quick cadence). To support continuous deployment, particularly across many delivery teams, you will need a large number of release slots. The implication is that you will also likely need more release windows more often. The advantage of quick cadence release windows is that they are less likely to suffer from the bottleneck challenges associated with slow cadence release windows and release trains.

Continuous release availability. With this approach delivery teams are allowed to release their solutions into production whenever they need to. In many ways this is simply an extension of the release window strategy to be 24/7.

This is the only strategy that truly supports continuous delivery. To make it work a host of DevOps practices are required, such as fully automated deployment, fully automated regression testing, feature toggles, self-recovering components, and many others are required.

- ✓ Shortening your time to value means your customers get a return on the time, money, or effort that they **invested much more quickly** so they will more likely stick with you than move elsewhere.
- ✓ Throughput is the **amount of a product or service that a company can produce and deliver to a client within a specified period of time**. The term is often used in the context of a company's rate of production or the speed at which something is processed.
- ✓ Agility means **effective (rapid and adaptive) response to change, effective communication among all stockholder**.
- ✓ is taking steps to reduce the risk (the **severity of the impact and/or probability of the occurrence**).

Benefits of Release Management



Release Management Tools

