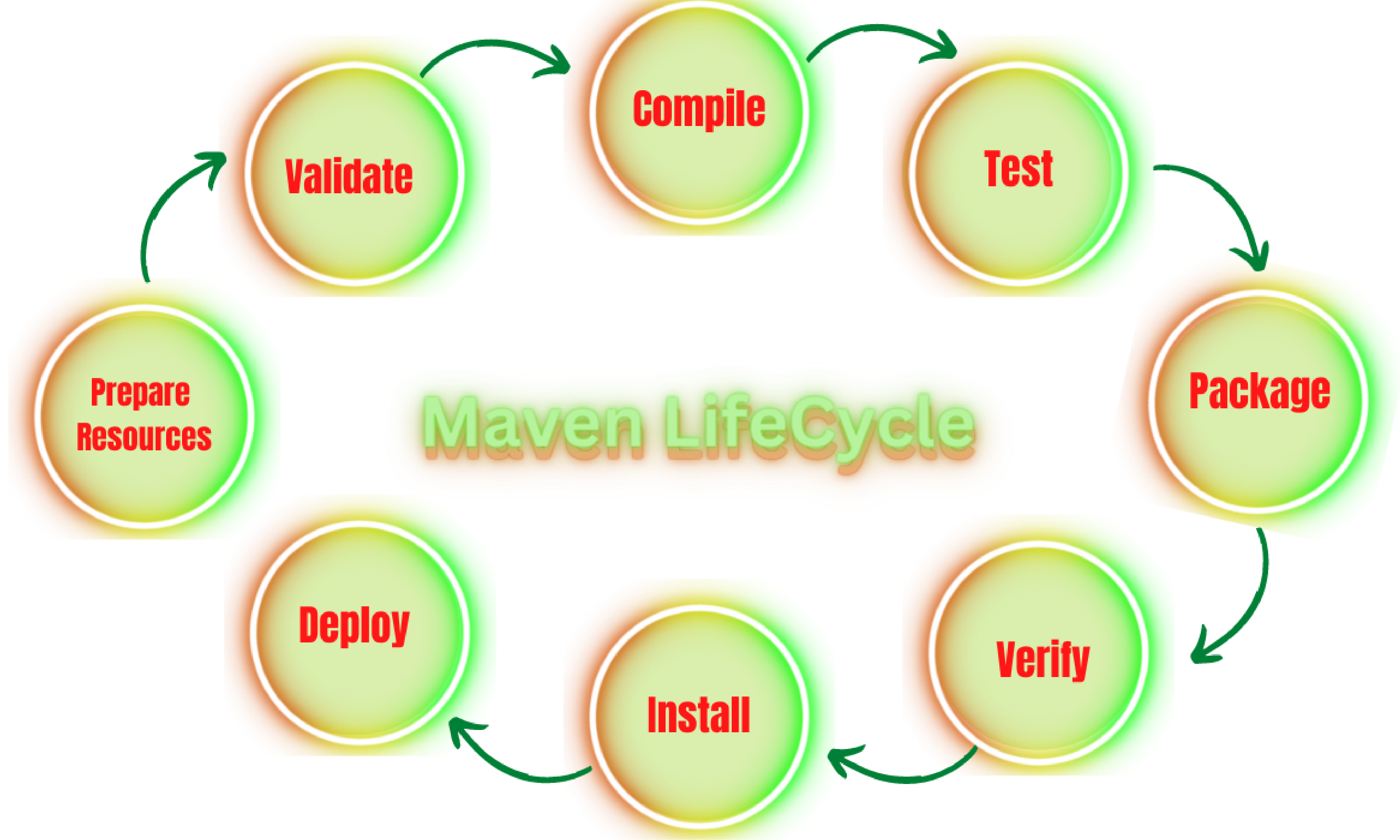


Introduction to Maven

- Maven is a powerful and widely-used build automation and project management tool.
- Developed by Apache Software Foundation, Maven simplifies the build process, making it more manageable and efficient.
- It provides a comprehensive infrastructure for managing projects, handling dependencies, and automating the build lifecycle.

Maven LifeCycle



Maven Goals

- **Clean:** Deletes the target directory.
- **Compile:** Compiles the source code.
- **Test:** Runs unit tests.
- **Package:** Creates a distributable package.
- **Install:** Installs the package in the local repository.
- **Deploy:** Copies the final package to the remote repository.

Advantages of Maven

- **Consistency:** Maven provides a consistent and standardized build process across projects.
- **Dependency Management:** Efficiently manages project dependencies, reducing manual effort.
- **Plugins:** Extensible through a wide range of plugins, allowing customization and integration with other tools.
- **Convention over Configuration:** Follows the principle of convention over configuration, reducing the need for explicit configuration.
- **Centralized Repository:** Maven central repository facilitates the sharing of dependencies across projects.

Maven Usage

- **Integration with IDEs:** Easily integrates with popular Integrated Development Environments (IDEs) like Eclipse, IntelliJ, and NetBeans.
- **Command Line Interface:** Allows for command-line usage, making it suitable for automated builds and continuous integration.
- **Build Profiles:** Supports the creation of build profiles, enabling the customization of builds for different environments.
- **Community Support:** Maven has a vibrant community, providing resources, plugins, and support through forums and documentation.