|  |  |  |  |
| --- | --- | --- | --- |
| **Artifact** | **What is it** | **Why it is used** | **How is it created** |
| Codebase | The collection of source code files that form the software system. | It is the foundation of the software, containing all the instructions and logic for the system. | Programmers and development tools write and contribute to the codebase. |
| Core Executable | The main executable file or files that constitute the functional software application. | It is the executable form of the software that users interact with. | Created through compilation of the source code using a build process. |
| Commit | A commit represents a specific version of the codebase, including changes made by developers. | Keeps track of changes made by contributors, allowing collaboration and version control. | Developers make commits using version control tools (e.g., Git), describing changes made. |
| Workflow | A series of automated steps that define the process of building, testing, and deploying the software. | Ensures consistency and efficiency in the software development lifecycle. | Defined in workflow configuration files (e.g., GitHub Actions YAML files). |
| Unit of Work | A set of related tasks or activities that need to be completed to achieve a specific goal. | Manages and tracks progress on a particular feature or bug fix. | Typically tracked in project management tools or issue trackers, associated with specific issues or tasks. |
| Readme | A document providing information about the software, its usage, and other important details. | Serves as documentation for users and developers, helping them understand the project. | Created and maintained by developers, often written in Markdown format. |
| Activity | Records of user interactions and system events within the software project | Provides insights into how users engage with the system. | Generated automatically by tracking user actions or events in the application. |
| Custom Properties | Additional properties or configuration settings specific to the software project. | Allows customization and configuration based on project-specific requirements. | Defined and managed by developers, often in configuration files or settings. |
| Actions | Automated tasks or processes triggered by specific events in the software development lifecycle. | Streamlines repetitive tasks, such as building, testing, and deploying code. | Defined in configuration files (e.g., GitHub Actions YAML files) and associated with specific events. |
| Wiki | A collaborative documentation space where project-related information is stored and shared. | Provides a central location for detailed project documentation. | Edited and updated by project contributors, often using wiki markup or Markdown format. |
| Tags | Identifiers assigned to specific points in the commit history to mark important releases or milestones. | Helps easily reference and navigate to specific points in the project history. | Created by developers using version control tools, typically associated with releases or significant changes. |
| Branches | Independent lines of development within the codebase, allowing for parallel work on different features. | Facilitates concurrent development and experimentation with new features. | Created and managed by developers using version control tools. |
| Pull Request | A proposed set of changes submitted by a contributor for review and integration into the codebase. | Facilitates code review, discussion, and collaboration among team members. | Created by contributors when they want to merge their changes into the main codebase. |
| Releases | Specific versions of the software that are considered stable and ready for deployment. | Indicates significant milestones or improvements in the software. | Tagged in the version control system and associated with specific commit states. |
|  |  |  |  |
|  |  |  |  |