

---

# Overview

## 1. Overview

A [Software Development Process](#)<sup>1</sup> captured from historical industry practices and evolving modern approaches. It's maintained as change is encountered and experienced in the IT industry.

**The purpose of the MDP is to encourage and support developers by:**

- Separating Development from Operations
- Identifying Automation separately from Development and Operations

**The goals of this process include:**

- Get more value to the market quicker
- Reduce risk
- Empower developers with ability to present all possibilities
- Promote an environment where developers can support any dx group
- Identify, develop, and maintain “Best Practices”

This process utilizes the [GitFork Workflow](#)<sup>2</sup> which is built on the [GitFlow Workflow](#)<sup>3</sup>.

---

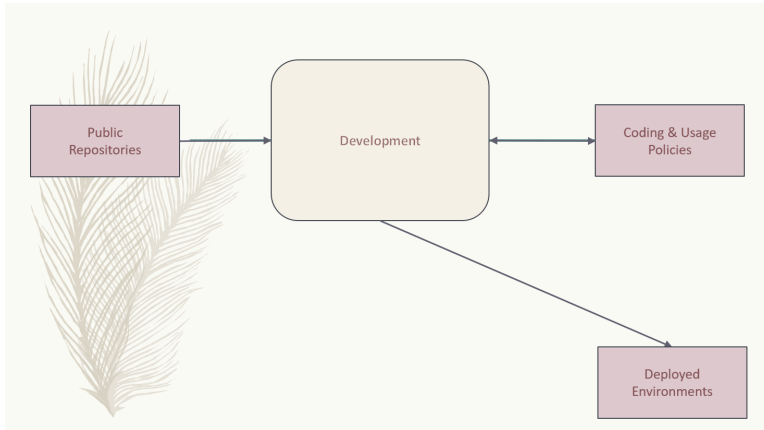
<sup>1</sup> [https://en.wikipedia.org/wiki/Software\\_development\\_process](https://en.wikipedia.org/wiki/Software_development_process)

<sup>2</sup> <https://www.atlassian.com/git/tutorials/comparing-workflows/forking-workflow>

<sup>3</sup> <https://datasift.github.io/gitflow/IntroducingGitFlow.html>

This approach has been defined and proven in the open source development world.

## 2. Context Diagram



### 2.1. Develop Changes

Developers are focused on development separately from Operations and CI<sup>4</sup> / CD<sup>5</sup> automation. They work directly with **public repositories** and local code identifying best options for development requirements or “asks”.

During this DevOps process, development references and challenges **Coding & Usage Policies** per security, licensing, testing, and coding standards.

Development proceeds unencumbered with private, individual developer controlled forked repositories, separate from the automation of CI/CD and stability required for operations.

Developers are free and encouraged to expand and improve CI/CD automation. This includes building, testing, and operational environment challenges as they develop new functionality for customer solutions.

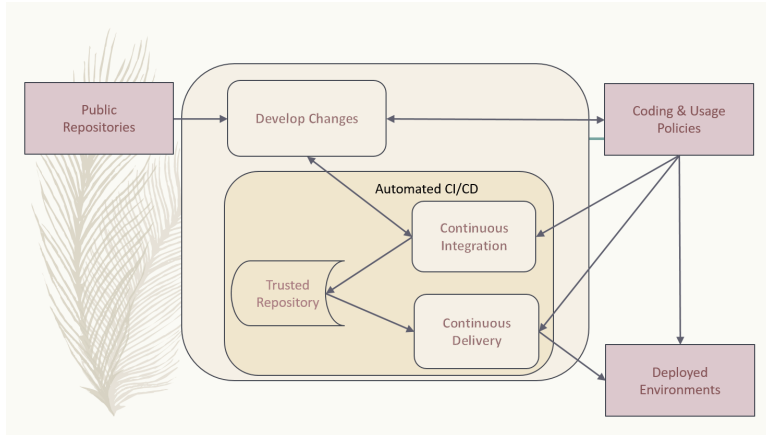
**Code Reviews** occur with Development Leads prior to initiating the Automated CI/CD.

---

<sup>4</sup> [https://en.wikipedia.org/wiki/Continuous\\_integration](https://en.wikipedia.org/wiki/Continuous_integration)

<sup>5</sup> [https://en.wikipedia.org/wiki/Continuous\\_delivery](https://en.wikipedia.org/wiki/Continuous_delivery)

### 3. DevOps Process



#### NOTES:

- For the [GitFork workflow](#)<sup>6</sup>, Dev coding, testing, verifying, etc, happens between step 7 and 8.
- **Rebuild project/repository OFTEN. All repos including workareas are:**
  - less stale
  - Up to date
  - More secure
- Project or repositories that are used by automated **Continuous Integration** are configured to promote code reviews and clean development baseline starting points. These are the repositories developers fork from.
- Developer forked repos can utilize CI and CD together to promote to a developer controlled environment. These type of pipelines are limited to a maximum set of versions in the **Trusted Repository** as they support development. Otherwise non-development pipelines end at the **Trusted Repository**.

<sup>6</sup> <https://www.atlassian.com/git/tutorials/comparing-workflows/forking-workflow>

### 3.1. Continuous Integration

An automated process that should be initiated at the completion of a **git push** to the primary (non-forked) GHE repository **develop** branch.

The build of this process enforces and verifies the **Coding & Usage Policies**.

**Provides tested deployment sets to the Trusted Repository:**

- All known tests, like code, are executed for every build prior to adding to Trusted Repo.
- This allows deployment environments, promoted to by **Continuous Delivery** to be focused on validation of new changes.
- Successful build identified with [semantic versioning](#)<sup>7</sup>.



**NOTE:**

- Changes to this process are developed and tested in the **Develop Changes** process, subject to code reviews.

### 3.2. Trusted Repository

This is the end point of the **Continuous Integration** process and the starting point of the **Continuous Delivery** process.

In addition to updates made by the **Continuous Integration** process, **Coding & Usage Policies** can report on and remove elements as required.

### 3.3. Continuous Delivery

Provides deployments from the **Trusted Repository** for argument selected environments.

**Coding & Usage Policies** can report on and remove elements as required within this process.

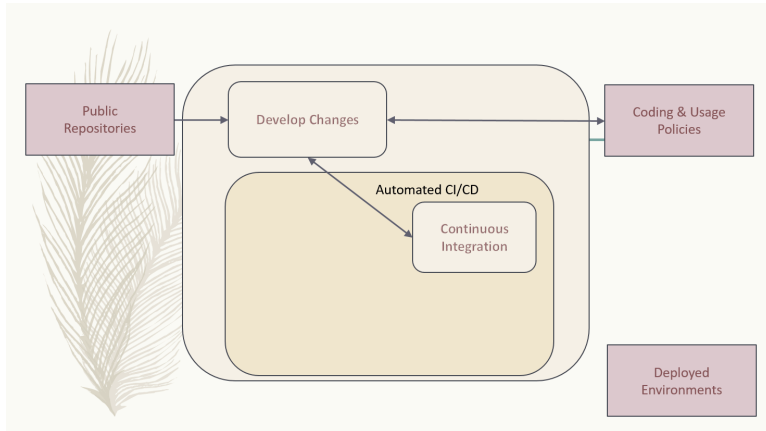
---

<sup>7</sup> <https://semver.org/>

**NOTE:**

- **Coding & Usage Policies** can report on and remove elements as required outside of the EMM DevOps Process.

## 4. DevOps Developer

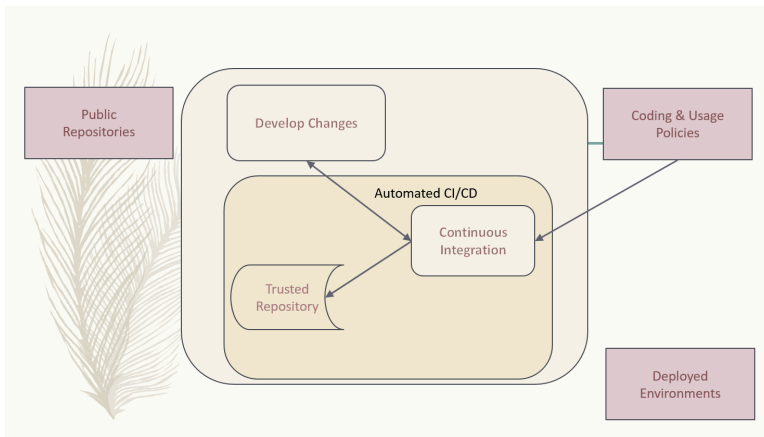


**For given “asks”, developer looks for and/or develops working solutions**

- Identify/Establish repository or repo:
  - From Public Repository or Team Repository or create new (Team repos available to CI process)
- Fork from existing projects or repositories for full development freedom:
  - Forking Projects – (example using the spoon-knife project),
  - Fork & Pull Workflow (For git beginners)
- Clone from forked repo
- Create additional remote to non-forked repo
  - NOTE: Rebuild local repo often, daily to start
- Verify Coding & Usage Policies:
  - Report variance to customers
  - Challenge security, licensing, testing, and coding standards as needed

- Prepare Changes:
  - Test with all known CI process testing. Add, modify, remove as needed.
- Commit and merge per GitFork workflow (Dev coding, testing, verifying, etc, happens between step 7 and 8)
- Initiate automated CI
  - Push Changes to Non-Forked repository or project

## 5. Continuous Integration Automated Process



**The Continuous Integration process is:**

- Initiated by Developer with a repository push
- Automatically verified with existing “Coding & Usage Policies”
- Automatic Build
- Automatic Testing
- Build/Test reporting to development
- Successful build identified with [semantic versioning](https://semver.org/)<sup>8</sup>
- Successful build delivered to Trusted Repository

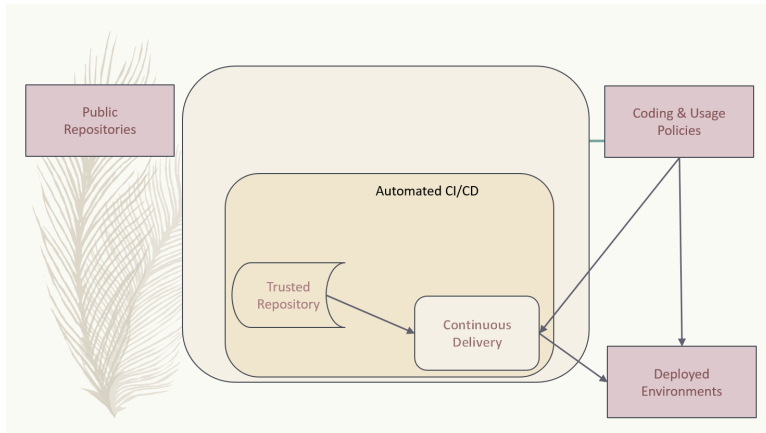
---

<sup>8</sup> <https://semver.org/>

**NOTE:**

**Coding & Usage Policies** verification reporting is run against all elements in Trusted Repository on a regular basis.

## 6. Continuous Delivery Automated Process

**The Continuous Delivery process:**

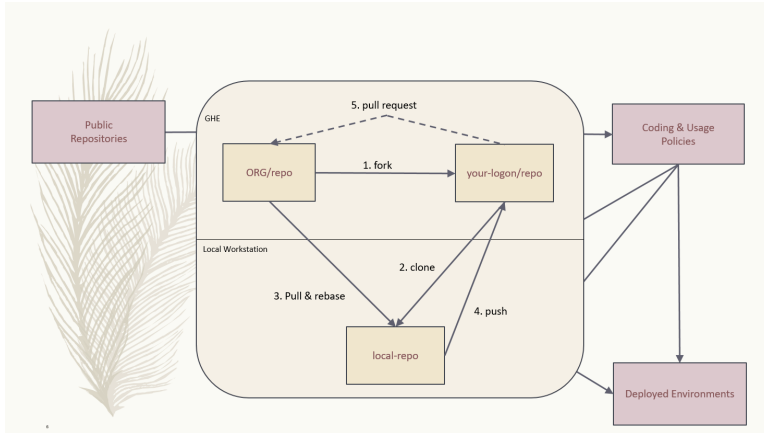
- Starts with up-to-date, verified, built from a Trusted Repository
- Contents included are delivered from CI automated process exclusively
- Uses Builds identified with [semantic versioning](https://semver.org/)<sup>9</sup>
- Automatically verified with existing “Coding & Usage Policies”
- Automatic environment deployment report
- Automatically deploys to specified environment

**NOTE:**

**Coding & Usage Policies** verification reporting is run against all "deployed to" environments on a regular basis.

<sup>9</sup> <https://semver.org/>

## 7. Fork Related Steps



### NOTES:

- Developers may have multiple instances of the above development environment depending on the projects/repos they support.
- Code reviews initiated by contributing developers and managed by DevLeads are the gateway to controlled GHE repository updates and non-development CI initiation.
- The use of forked repos only requires ready-only permission for contributing developers.
  - Executing the [Install git<sup>10</sup>](http://emmutl-ch2-1p.sys.comcast.net/git/EMM-GitInstallationProcedure.adoc.html) procedure is required for this.
- The Develop Changes process is separate from:
  - Automation:
    - CI
    - CD
  - Operations

<sup>10</sup> <http://emmutl-ch2-1p.sys.comcast.net/git/EMM-GitInstallationProcedure.adoc.html>



## 8. References

- [Fork & Pull Workflow](#)<sup>11</sup> For git beginners
- [Forking Projects](#)<sup>12</sup> – example using the [spoon-knife project](#)<sup>13</sup>
- [GitHub and Git Foundations \(YouTube\)](#)<sup>14</sup>

---

<sup>11</sup> <https://reflectoring.io/github-fork-and-pull/>

<sup>12</sup> <https://guides.github.com/activities/forking/>

<sup>13</sup> <https://github.com/octocat/Spoon-Knife>

<sup>14</sup> <https://www.youtube.com/playlist?list=PL0Io9MOBetEHhfG9vJzVCTiDYcbhAiEqL>

