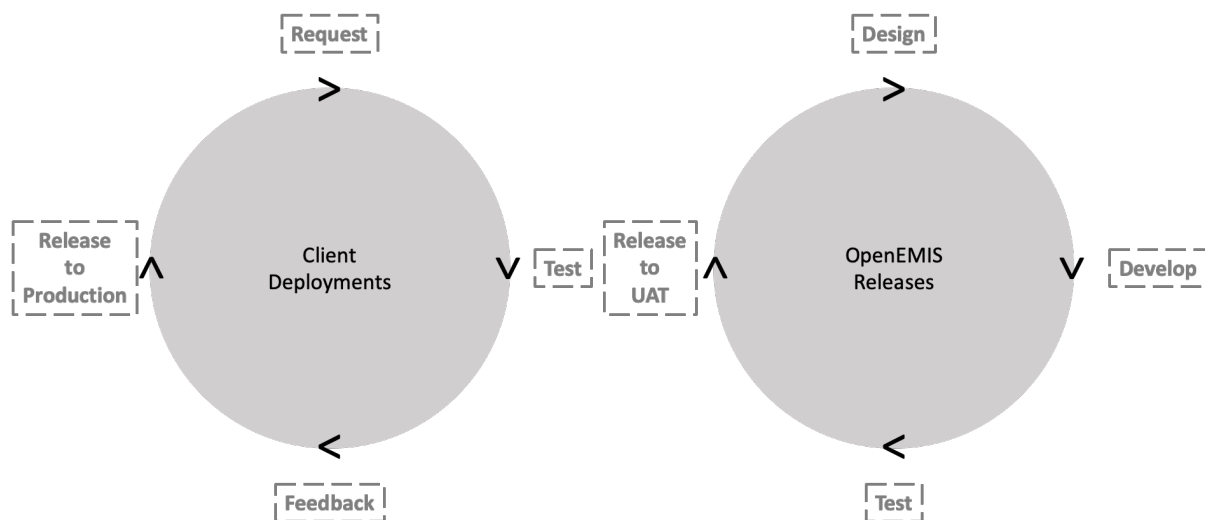


OpenEMIS Release Management

The OpenEMIS initiative has adopted the agile software development methodology and therefore there are typically a much higher number of releases. This approach is often referred to as Continuous Delivery. The goal of Continuous Delivery is to release applications more frequently. The movement of code from development to production is referred to as a “release”. The OpenEMIS initiative uses the ITIL framework to manage the release management process. The ITIL release management process "aims to plan, schedule and control the movement of releases to test and live environments".



The terms "version release" and "version deployment" are closely related in the context of software development, but they have distinct meanings.

Release:

A **release** refers to the act of making a specific version of a software application available to users. It involves making the software package, which includes the compiled code, documentation, and any other necessary files, accessible to end-users for installation or download.

Deployment:

On the other hand, **deployment** refers to the process of taking a **release** of a software application and making it operational in each environment. This involves not only making the software available but also ensuring that it runs smoothly in the target environment.

Before doing a **deployment** to production, it is highly recommended to do a **deployment** to a UAT (user acceptance testing) environment to identify any potential issues. This ensures that the deployment process goes smoothly and that the new version behaves as expected.

In summary, a release is the point at which users can obtain a new version of the software, while deployment is the point at which users can use a new version of the software.

In the realm of software development and maintenance, deploying updates is a critical aspect to enhance functionality, fix issues, and improve overall user experience. There are various types of deployments, each serving specific purposes. This document aims to clarify the distinctions between Version Deployment, Data Patch Deployment, and Hotfix Deployment.

The OpenEMIS team is committed to delivering regular updates to enhance user software experience. We understand that testing and deploying every new version can be resource-intensive for your organization. To help come up with a deployment policy, we have outlined some guidelines that you may find useful:

1. Major vs. Minor Versions (see below):
 - Consider thorough testing for major releases due to significant changes.
 - Streamline testing for minor releases, focusing on bug fixes.
2. Change Log:
 - Review change log to understand the impact of each update.
3. Testing Environments:
 - Maintain testing environments that mirror your production setup for accurate testing.
4. User Acceptance Testing (UAT):
 - Involve schools in testing for major version using real-world data.
5. Scheduled Maintenance Windows:
 - Establish scheduled maintenance windows for planned testing and deployment.
6. Risk-Based Testing:
 - Prioritize testing efforts based on the impact of introduced changes.
7. Testing Frequency:
 - Deploying versions too quickly can create a stress on the testing team and overwhelm schools and create an unstable environment
 - Deploying versions too slowly means that clients miss out of fixes and feature and could result in unsupported version
8. Rollback Plan:
 - Have a well-defined rollback plan in place to minimize downtime in case of unexpected issues.
9. Communication Channels:
 - Immediately report any critical releases via the service desk for follow up.
10. Training and Communication:
 - Ensure that schools are aware of the changes and provide resources to help users adapt to new features.

We believe these guidelines will help you optimize your testing and deployment processes.

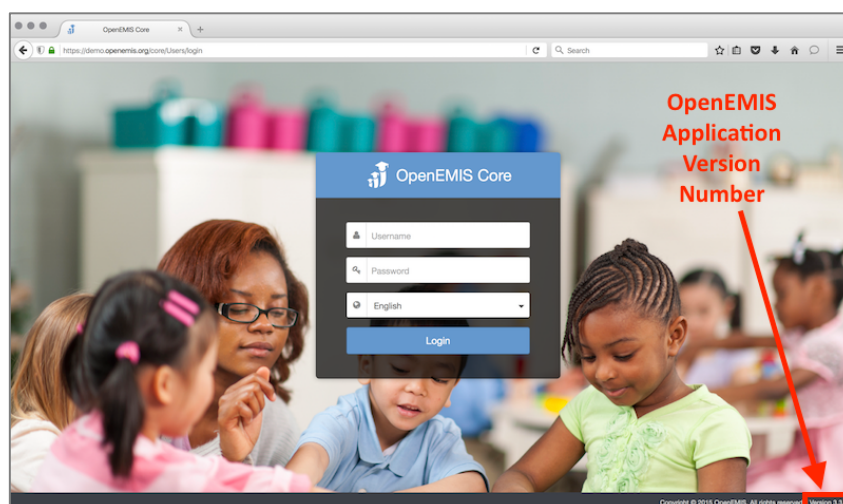
Version Deployment

A **Version Deployment** involves releasing a new version of the software application.

Attributes of the software application	Changes
Data	No
Code	Yes
Database	Yes

All OpenEMIS software versions have three (3) numbers (eg v4.1.2).

- First number: Major Enhancements
- Second number: Minor Enhancements
- Third number: Security / Bug fixes



Each new version goes through multiple QA (Quality Assurance) test phases:

- Unit testing
- Manual / Functional testing
- Automation testing
- User testing

The detail of each new version is added to the change log:

<https://demo.openemis.org/core/CHANGELOG.md>

Clients can then test new version in UAT prior to deploying new versions to production.

Any questions or feedback from clients regarding the latest proposed release can be logged via the respective OpenEMIS Service Desk.

Data Patch Deployment

A **Data Patch Deployment** is focused on updating or modifying the data within the system, without changing the underlying code or application version.

Attributes of the software application	Changes
Data	Yes
Code	No
Database	No

The OpenEMIS may from time to time release data patches to fix data issues in the database. Again these data patches should first be deployed and tested in a UAT environment before deploying to production.

Hotfix Deployment:

A **Hotfix Deployment** is a small, targeted update to address a specific critical issue or bug in the software without introducing new features. Hotfixes are a subcomponent of a **Version Deployment**. The **Version Deployment** contains fixes for multiple issues whereas the **Hotfix Deployment** contains the fixes for a single issue.

Attributes of the software application	Changes
Data	No
Code	Yes
Database	No

WARNING: there is a risk in **Hotfix Deployment** as the files may not be compatible with the existing **Version Deployment**. The practice **Hotfix Deployment** does not constitute an official **Version Release** as it has not been tested or verified by the OpenEMIS lab. Ideally, the best solution is to deploy an official tested and certified **Version Release**.