



Lesson Objectives

- Introduction To Software Versioning
- Major Release
- Minor Release
- Revision Release
- Build Release
- Beta Version For User Testing



8.1: Introduction To Software Versioning

Introduction To Software Versioning



- Software versioning procedures fall into the category of a business process required to be clearly defined within an organization.
- Without a well-defined process it becomes increasingly difficult for an organization to support, maintain, trace and account for the applications that are released to its customers and to identify customers that need to be notified when updates or upgrades are available for them, specifically updates that improve system behaviours.
- **Software** - means machine-readable instructions and data (and copies thereof) including middleware and firmware.
- **Version** - is a state of an object or concept that varies from its previous state or condition. The term "version" is usually used in the context of computer software, in which the version of the software product changes with each modification in the software, the "version" is a specific "revision" of the software.

8.1: Introduction To Software Versioning



Introduction To Software Versioning

- Versions and Releases are denoted using a quadruplet of integers to indicate Major, Minor, and Revision (Rev); Build numbers are for internal tracking and verification of the software build process and will not be visible to customers are part of the software version number.
- The rationale and rules to be used in assigning a version number are addressed later in this document.
- <MAJOR>. <MINOR>. <REVISION>.<BUILD>

8.1: Introduction To Software Versioning

Introduction To Software Versioning



Release 6.2.0.0

Major version
Minor version
Maintenance version
Patch version

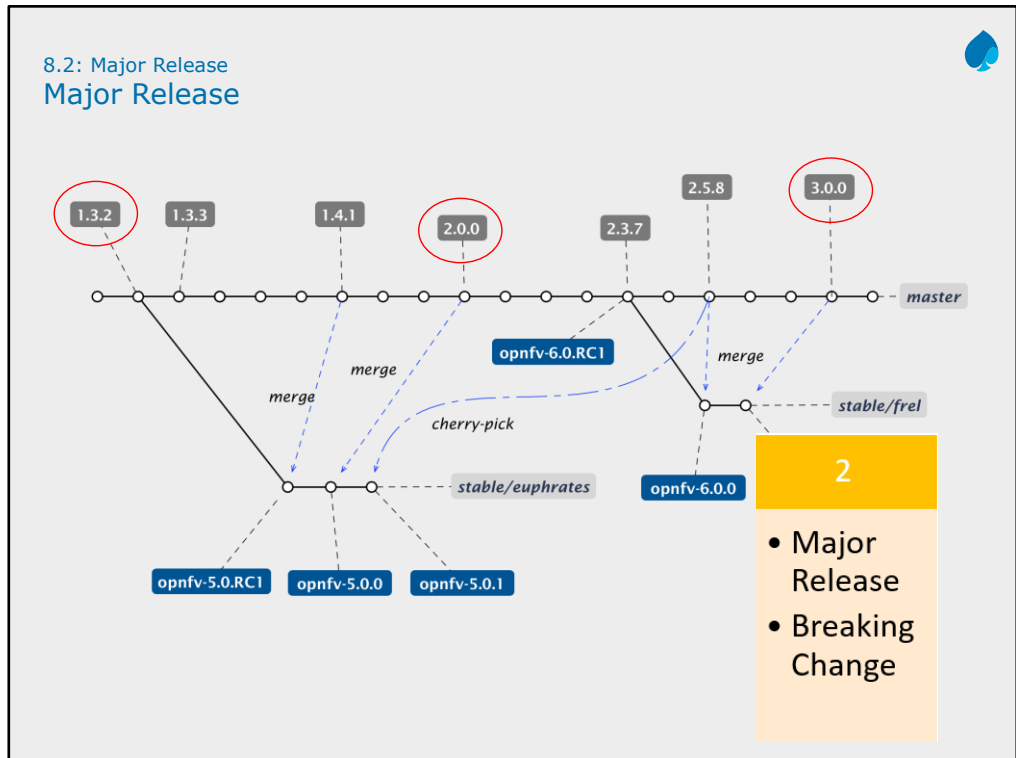
Examples
Release 6.2.0.0
Release 6.1.0
Release 5.2.2
Release 6.0
Release 6.1.x
Release 6.x

8.2: Major Release

Major Release



- A Major Release is a full product release of the software. It generally contains new customer-facing functionality and represents a significant change to the code base comprising the software product or family of products, or is used to represent a significant marketing change or direction in the product.
- Major version numbers are generally incremented by the product management team, and generally are accompanied with a new marketing push, or to communicate a significant improvement to the product.
- The major version number is identified by the digit or set of digits to the left of the decimal point (e.g., x.0, segment 1 with a placeholder value of x).



8.2: Major Release

Major Release

**General:**

- The major version number will be unique for system feature sets; a major release may contain minor, release candidate, and/or build modifications of the major release.
- May be accompanied by a branch of the code base within version control for further minor and/or release candidate development.

Scope:

- Any change to the code base that prevents backwards compatibility (e.g. Addition or Removal of features, changes in the DB schema or API commands).
- Any new functionality that is customer facing.
- Any large marketing push that accompanies the product and redirects the product.

Example:

- 2.x, 3.x, etc...

Frequency:

- Market driven

Audience:

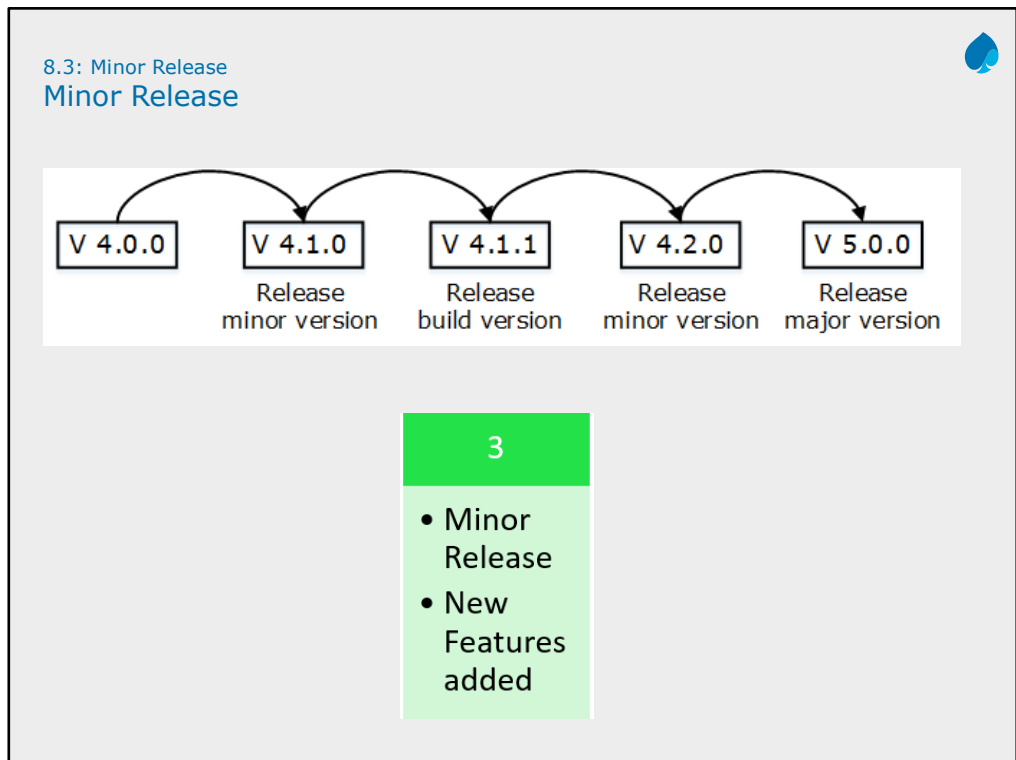
- New customers.
- Existing customers with qualifying contracts
- Existing customers desiring to upgrade to the new feature set

8.3: Minor Release

Minor Release



- A Minor Release of the software may be comprised a rollup of several branched releases, enhancements/extensions to existing features or interfaces driven by internal or external requirements, external requirements could be driven by enhancements to meet new sales area (State specific features or rules), internal requirements could be enhancements aligned to a new marketing push.
- The minor version number is identified by the digit or set of digits to the right of the decimal point separating it from the Major Release number (e.g., 2.x, x indicates the Minor Release placeholder).



8.3: Minor Release

Minor Release

**General:**

- The minor version number should always be unique, unless the release is a Revision or a Build Release.
- Must be accompanied by a label in version control.
- May be branched in version control for further patch development.
- Any minor releases of a particular major release are considered to be a part of the latest major release (e.g. 2.4, 2.5 and 2.6 are all part of major version 2.0).

Scope:

- Minor enhancements and features that do not affect compatibility with its associated major or current minor releases.
- New features or functionality that does minimally effect the interfaces
- Addition of new features or functions to meet a new sales area that doesn't affect existing customers of the release.
- To accompany or communicate a new marketing initiative.
- Error corrections and maintenance

8.3: Minor Release

Minor Release

**Example:**

- 2.4, 2.5, etc...

Frequency:

- Enhancement and Market driven

Audience:

- New customers.
- Existing customers with qualifying contracts.
Existing customers desiring to upgrade to the new feature set

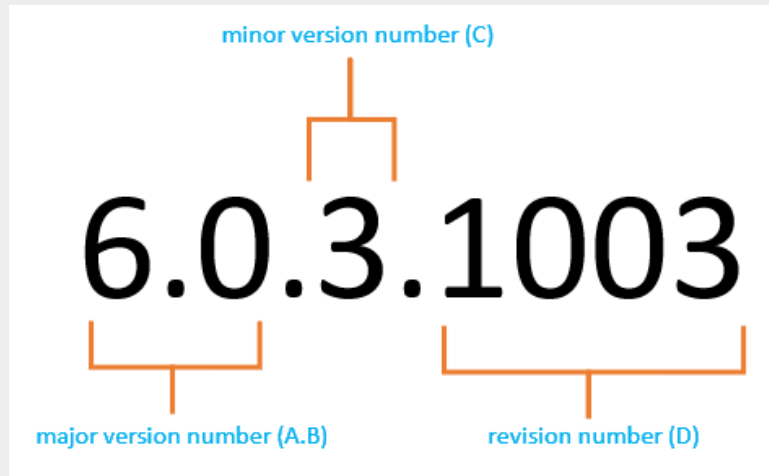
8.4: Revision Release

Revision Release



- A Revision (Rev) is a build of all or part of the software that is initially distributed to an internal audience, specifically software quality assurance, for software validation.
- If the Rev is successfully validated and accepted, this version is "released" to manufacturing.
- If defects are found that prevent the Rev from successfully being validated and prevent the release to manufacturing, the Rev value will be incremented prior to the next validation cycle.

8.4: Revision Release Revision Release



8.4: Revision Release

Revision Release

**General:**

- The Rev version number should always be unique, unless the build is a SVN build of the original Rev release.
- Any Rev build of a particular major or minor release are considered to be a part of the latest minor release (e.g. 2.4.5, 2.4.6 and 2.4.7 are all part of minor version 2.4 and major version 2.0).
- May be branched in version control if SVN builds are needed.

Scope:

- Releasing build to SQA for validation

Example:

- 2.4.1, 2.4.2, etc...

Frequency:

- As necessary, but typically every 1 to 3 months.

Audience:

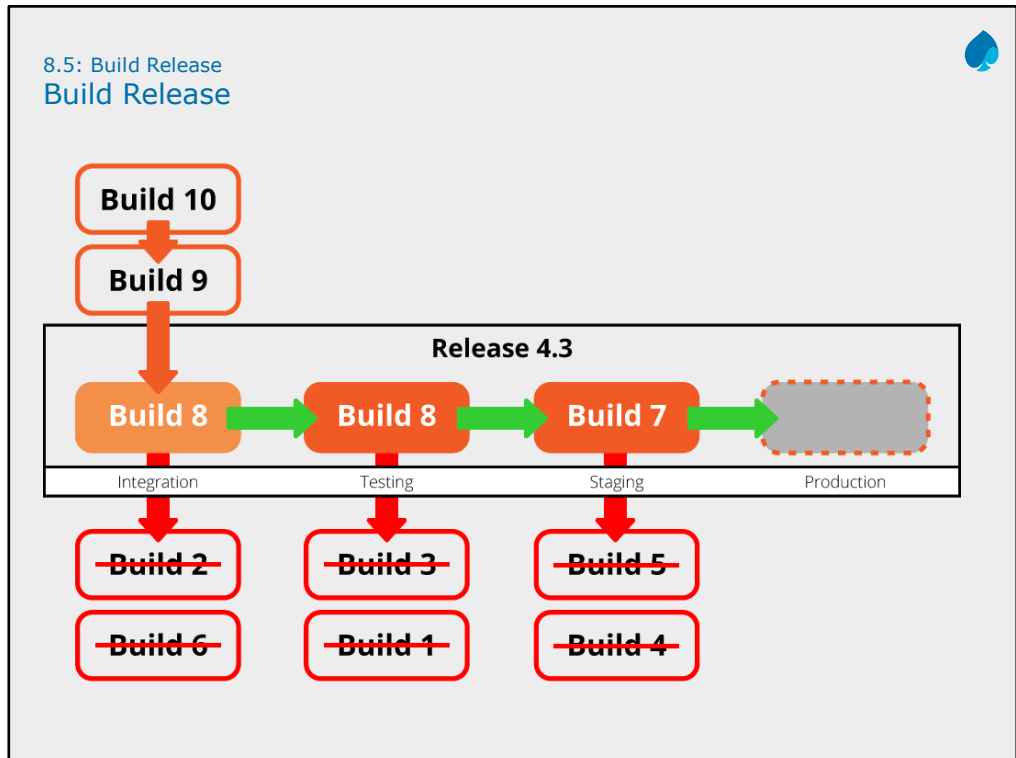
- Existing customers with qualifying contracts.

8.5: Build Release

Build Release



- A Build Release is a build of all or part of the software distributed to an internal audience, these releases should have targeted feature enhancements and issue resolution documented to allow testing in the targeted/specific areas where the changes were implemented.
- The Build version number is denoted by a fourth digit or set of digits (e.g. 2.4.7.x, segment 4 with a placeholder value of x), corresponding to an internal build number.



8.5: Build Release

Build Release

**General:**

- The Build version number must always be unique.
- Any Build of a particular major, minor or patch release are considered to be a part of the latest Rev release (e.g. 2.4.5.1 is part of patch version 2.4.5, minor version 2.4 and major version 2.0).

Scope:

- Resolves a particular defect, typically of a critical Severity level with no viable workaround.

Example:

- 2.4.7.1, 2.4.7.2, etc...

Frequency:

- Extensively used during internal developement.

Audience:

- Internal developers
- Internal verification

8.6: Beta Version For User Testing

Beta Version For User Testing



- Beta version is made available for testing during the software development.
- It is only released to a limited number of users before it can be released for general.
- Beta software is released mainly to help the developers test the software performance, compatibility, and stability while identifying problems such as bugs, lags, or crashes.
- For users, a Beta software is a trial version of the official software that they can download to try out the new features and improvements of existing features that will be incorporated into the official release.



Summary



In this lesson, you have learnt

- In this lesson, you have understood the different Software Versioning
- Major Release
- Minor Release
- Revision Release
- Build Release
- Beta Testing



Add the notes here.

Review Question



Question 1:

Which release version number must be unique?

- Major Release
- Minor Release
- Revision Release
- Build Release



Question 2: True/False

- Beta Testing is released to all the users before its general releasing.

Question 3: Fill in the Blanks

- _____ is a full product release of the software.