

TERMINOLOGY OF PRODUCT RELEASE

This document serves as a common vocabulary to discuss product release and delivery packages in the product lifecycle.

MOCKUP

A visual presentation of a design or page layout that approximates what the final printed piece might look like. A mock-up should not be confused with a prototype, defined in this document.

Prototypes are always meant to function, even if not fully so, whereas mock-ups are only meant to look like the real system, and do not function. The two are easy to confuse because mock-ups are a way of prototyping user interfaces on paper or computer images.

CHARACTERISTICS

- Does not perform significant functionality of any kind
- Focuses on the visual experience of a proposed product or feature
- Typically generated by the User Experience Team as part of a project's envisioning phase
- Not intended to be released to clients.
- Used by product managers to ensure needed functionality in product and dependent systems
- Used by architecture and engineering teams to guide development

REFERENCE IMPLEMENTATION

A reference implementation is an example of a standard for use in helping others implement versions of the standard. A standard is much easier to understand with a working example in hand. The purpose of a reference implementation is generally to increase awareness and familiarization of the spec within the development community.

While it is entirely possible for reference implementation (RI) software to serve in the academic cause of pure knowledge, on a more pragmatic level they are generally intended to familiarize developers and the market with a specification so that developers will be more likely to develop commercial implementations and users will be more likely to purchase conforming implementations.

A small example would be an XML document and Schema published along with a policy detailing how XML documents will implement validation rules via XSD schema.

CHARACTERISTICS

- Typically generated by a software architect
- Typically accompanies a formal policy or recommendation
- Used by software engineers as an example of how to solve a problem



- Potentially used by customers as an example of how a problem has already been solved

PROTOTYPE

A prototype is a rapidly developed working model used to test various aspects of a design, illustrate ideas or features and gather early user feedback.

When the prototype is sufficiently refined and meets the functionality, robustness, manufacturability and other design goals, the product is ready for implementation in conformance with existing standards of quality and support.

The process of prototyping involves the following steps

1. Identify basic requirements
2. Develop initial prototype
3. Review
Customers, including end-users, examine the prototype and provide feedback on additions or changes.
4. Revise and Enhancing the Prototype
Using the feedback both the specifications and the prototype can be improved. If changes are introduced, the prototyping team may repeat steps #3 and #4.

CHARACTERISTICS

- Typically generated by a software architect or software development team
- Functionality driven by functional requirements or feature definitions
- Shown to customers as a means of generating client feedback for product evolution
- Does not typically solve for the operational constraints of a system
 - Security
 - Extensibility
 - Traceability
 - Monitoring
 - Etc.
- May become part of a formal specification to a product development team

CTP – COMMUNITY TECHNOLOGY PREVIEW

Occasionally a build of a product is issued before alpha or beta versions in order to demonstrate current thinking of a product's development direction to technical customers. The primary difference between a CTP and a prototype is that a CTP is a release of a product from its actual development team. The intent of issuing a CTP is more to inform a client rather than to solicit feedback that may alter product direction.

CHARACTERISTICS

- Typically targeted at technical customers or technical staff within client organizations



- May be used as an early warning system to assure notification is made to the client
- Is not feature complete
- Created by the product development team
- Delivered with no guarantees of upgradeability

ALPHA

An Alpha delivery of a product is intended to be delivered to trusted testing staff or partners. Although an alpha delivery is feature complete, the quality of the delivery is not guaranteed. The purpose of delivering an Alpha release is to get testing on the product.

CHARACTERISTICS

- Has all features implemented targeted to a particular release
- Built and delivered through production systems and processes or by the product development team
- May be used as an early warning system to assure notification is made to the client
- Does not have sufficient testing done to assure stability
- Created by the product development team
- Delivered with no guarantees of upgradeability

BETA

This is often the first release of software generally available to clients or the public. Beta level software generally includes all features, but may also include known issues and bugs of a less serious variety. Beta releases test the supportability of the product, the go-to-market messaging (while recruiting Beta customers), the manufacturability of the product, and the overall channel flow or channel reach.

Beta version software is likely to be useful for internal demonstrations and previews to customers, but is considered unstable and not yet ready for release.

CHARACTERISTICS

- Built and delivered through production systems and processes
- Intended to install and run in real world environments
- Typically released as part of a managed program of customer feedback
- Typically accompanied by a list of known issues
- Has all features implemented targeted to a particular release
- May be used as an early warning system to assure notification is made to the client
- Sometimes delivered with guarantees of upgradeability

RELEASE CANDIDATE



The term release candidate refers to a version with potential to be a final product, ready to release unless fatal bugs emerge. In this stage, the product features all designed functionalities and no known showstopper-class bugs. At this phase the product is usually code complete.

A release is called code complete when the development team agrees that no entirely new source code will be added to this release. There may still be source code changes to fix defects. There may still be changes to documentation and data files, and to the code for test cases or utilities. New code may be added in a future release.

CHARACTERISTICS

- Built and delivered through production systems and processes
- Intended to install and run in real world environments
- Typically released to provide early access to a product before mass manufacturing has occurred
- Has all features implemented targeted to a particular release
- Often delivered with guarantees of upgradeability

RELEASE

The gold or general availability release version is the final version of a particular product. It is typically almost identical to the final release candidate, with only last-minute bugs fixed. A release is considered to be very stable and relatively bug-free with a quality suitable for wide distribution and use by end users.

CHARACTERISTICS

- Available to the Sales Team to be sold
- Release is accompanied by complete marketing and sales messaging
- Support infrastructure is in place to manage the product in the field
- Built and delivered through production systems and processes
- Has all features implemented and verified