Software Application Release Best Practices

Brief Overview

Software as a Service applications are constantly evolving to continue to provide customers with new features and improvements to highly used existing features. These changes may be obvious to an end user or may be more subtle changes of the HTML to make the application work more efficiently.

These changes are generally handled by the application developer themselves, without action needed from the customer. However, when multiple software applications are connected at the customer level, the customer may need to perform regular software maintenance that coincides with releases. A common example of this is a database application that pushes and pulls data from multiple user-facing applications.

As WalkMe is built on top of these different user-facing applications, builders should be aware of some basic software release best practices to ensure that their WalkMe content functions in an optimal manner.

Understand Your Application Release Schedule

Most applications make updates via scheduled releases throughout the year and, on average, provide these updates to their customers a month in advance of the production release to ensure time for proper testing. These releases may be categorized differently depending on the application vendor, but they usually are one of the following:

1. **Major Releases (generally 2-3 times/year)**
   - Application vendors batch new feature releases and major user interface updates into large software updates. This allows them to make large interdependent changes that can fundamentally change how users interact with the software.
2. **Minor Releases (numerous)**
   - These are often small software tweaks that improve upon existing features or processes. Often they are released independently of the major releases if they significantly improve the user experience but are too far from the major release cycle.
3. **Software Patches (numerous)**
   - These are often due to bugs existing in the software that impair the user experience and the full functionality of the software. These are released as soon as possible to ensure customer satisfaction.

Understand Your Environment Hierarchy

Different environments are generally used to signify different levels of end user software readiness. At the most basic level, applications will have a production environment and a sandbox/preview environment.

- The production environment is the environment that is actually in use by the business and houses real data and triggers real business processes.
- The sandbox/preview environment should mirror production as closely as possible and is an area where updates are tested before being released into the production environment to avoid as many issues as possible.

In reality, there may be many different types of sandbox/preview environments for a application: some use real data or test data, some closely mirror production, and some differ significantly. Sandbox/preview environments can also have different release schedules which will affect your ability to test application updates. Due to this, it is important to understand your environment hierarchy so that you can ensure your WalkMe content will perform as desired in your production environment.

WalkMe recommends that you build and test your content on a sandbox/preview environment that is as close to the production environment as possible. For major application releases, it is recommended that you test on an environment that has the new updates and is as close a representation as possible to what the updates will look like in production.

Periodic Maintenance of Your WalkMe Content

Once you understand your application release schedule and your environment hierarchy, you can plan when and where to check your WalkMe content to ensure that it evolves alongside your application. For quality assurance testing best practices, please check out [this article](#)!

WalkMe recommends you choose 30 of your most important WalkMe items (use Insights usage or business goals to prioritize) and play them on the current version of your application 1-2 weeks prior to a scheduled major release. Then, as soon as you have access to the upcoming release, play those same 30 items on the new version. This will let you better understand exactly what has changed in the newest release vs. what may have been a previously unnoticed defect.