

Estate Valuations & Pricing Systems, Inc.

Software Development Process

Overview

EVP Systems develops a suite of Microsoft Windows programs called EVP Office (consisting of EstateVal, CostBasis, CapWatch, and GiftVal), to retrieve publicly available financial data for estate and gift tax reporting purposes. The software connects via the Internet to the EVP Data Center, a Linux-based server application. Additionally, the company offers EVP Everywhere (a webservice) and EVP Upload (a file-based batch processor) that provide the same services, both of which are Linux-hosted.

Versions

EVP Office versions are defined by a three-part version number, per the Semantic Versioning specification (semver.org). The first number is a major release, usually coincident with new applications being added to EVP Office or because major portions of the existing applications have been re-written. The second number is a minor release, when new features have been added to the existing applications. The third number is a patch release, including bug fixes and minor improvements to the existing applications. If a “mid-release” version is required for a client, a fourth number is appended, to identify the single-client emergency patch.

The current release number of EVP Office, and the changes it contains, is available at www.evpsys.com/software. The full list of software changes, going back to 2001, is available at www.evpsys.com/software/release.

Life Cycle

EVP Systems uses a six-step software development life cycle.

During this process, no client data is used. EVP Systems does not have access to confidential client information, including personally identifiable information (PII) or Sensitive Personal Information (SPI), and so it cannot be involved.

Record / Classify

Suggestions for enhancements and fixes for bugs are collected by EVP Systems’ Support Department from both internal and external sources, and categorized by management according to their urgency, complexity, and client impact.

Assess

Development, Support, and EVP Systems' management meet on a regular basis to discuss the costs, risks, and benefits of each requested change. Special attention is given to the security implications of any changes.

Small fixes and features are usually approved for the next patch release.

Any change that modifies the data items or calculations used by EVP Office, or how that data is stored or presented in reports, requires at least a minor release version.

New applications or major changes to existing applications are long-term strategic efforts, and require a major-version release.

Plan

For each pending release, a development plan is proposed, citing the expected changes to the software, along with the checks required to adequately test those changes. Management approval must be granted before the plan can be put into action.

Develop / Test

Development and testing are concurrent processes.

Source code is stored on, and versioned by git, via a private repo on `github.com`. git allows for multiple branches, change management, and rollback. A copy of the executables for the latest release as well as the full source tree is also stored on `rsync.net`.

Access to the source repo is tightly controlled, and accessible only by the company's senior management and Development Department. An automatic post is sent to the company chat server whenever changes are committed to the repository.

EVP Office is built with Microsoft Visual Studio 2022, on Windows 11. All code is compiled with Level 3 Warnings (`/W3`) and Treat Warnings as Errors (`/WX`) enabled, and is statically analyzed with `cppcheck`, with no errors or warnings. The installation wizard is compiled by Inno Setup (`www.jrsoftware.org`), and signed with a Comodo certificate by `kSign` (`codesigning.ksoftware.net`). A non-interactive installer is created from the installation wizard by `MSI Wrapper` (`www.exemsi.com`), also signed by the Comodo certificate.

The EVP Systems Data Center is built with GNU C, on Canonical Ubuntu 20.04. All code is compiled with All Warnings (`-Wall`) and Treat Warnings as Errors (`-Werror`) enabled, and is

statically analyzed with `cppcheck`, with no errors or warnings. The language standard used is `gnu11`, an extension of C11 (ISO/IEC 9899:2011).

The supporting infrastructure for EVP Everywhere and EVP Upload (outside of the core C code of EstateVal) and many EVP Systems internal tools are written in Python 3. All Python code used in production is passed through `flake8`, with no errors or warnings.

As features and fixes are built, tests are developed concurrently, and run when the change is declared complete. Before a Develop / Test cycle is complete, a full suite of regression tests are also run.

Internal Release

After a release candidate of EVP Office has completed development, and passed all unit and regression tests, it is distributed to EVP Systems' Professional Services and Support Departments, for internal use.

Releases of the EVP Systems Data Center, after passing unit and regression tests, are made available at an internal IP address, for use by the Professional Services and Support Departments.

EVP Everywhere, EVP Upload, and internal tools releases are made available at an internal IP address, for use by Professional Services and Support Departments.

For all systems, the resulting reports, Data Center logs, and outputs are reviewed for anomalies, and if any are found, the release is returned for further development and testing.

Beta versions of EVP Systems' software are never released externally. If a client has an emergency situation that requires a new release of the EVP Office applications, a "mid-release" version is created specifically for them, to only address the specific issue they're having.

Final Release

After a period of successful use by Professional Services and Support, the new version of EVP Office is released to the public, via the EVP Systems website (www.evpsys.com/software). A change history is added to the site, a post is made to the site's front page (depending on the scope and scale of the added features and fixes), and existing users are notified directly by a button that becomes visible in the existing versions of the applications. E-mail is also sent to clients who have requested notice that way.

EVP Systems' clients have the sole discretion about upgrading their copies of EVP Office, based on their own needs and internal processes. EVP Systems' does not have access to client

workstations, and does not have the capability to upgrade, push, or patch their copies of the software.

EVP Systems normally schedules an end-of-life for each version 30 to 36 months after its initial release. If a significant problem is discovered with a release, an Emergency Notification is sent and access to the EVP Systems Data Center may be disabled for that version.

Releases of the EVP Systems' Data Center, EVP Everywhere, and EVP Upload are made during regularly scheduled maintenance windows—Saturdays, between 9:00 pm and midnight, Pacific Time.

Releases of EVP Systems internal tools are made as-needed, though the preferred time is off-hours, when normal work will not be disrupted.

Emergency Notifications

In the event a significant problem is discovered with a released version of EVP Office, EVP Systems has the ability to both disable the release and return a message to clients who try to use it, and notify clients who have produced reports with the flawed version. The notifications can take the form of:

- An in-program alert via the EVP Office Message Center
- An e-mail or a phone call to the client's contact in EVP Systems' account system
- A posting on EVP Systems' web site

Though EVP Systems' does not have any personally identifiable information (PII) or Sensitive Personal Information (SPI) for the reports produced by EVP Office, it can provide the type of report that was produced, the time when it was evaluated, and which securities it included. If the portfolio was evaluated with "passthrough" data—unique identifiers provided by the user for a portfolio—that can also be added to the alert.

Last Update: October 13, 2024