Engineer

ソフトウェア

コンピューターで実行するプログラムと、それが扱うデータ

どのようなものか

ソフトウェアとは、はんだ付けではなく、書き換えることで変更できるコンピューターシステムのあらゆる部分を指します。プログラム、それが扱うデータ、動作を指示する設定が含まれます。英語では不可算名詞なので、単体を指すときは「a piece of software」や「a program」と言います。人が作業のために開くものは、通常アプリケーションと呼びます。「コード」は開発者が編集するソースを指すことが多く、仕様書では全体をSWと略すこともあります。

通常、その対になるのは物理的な機械であるハードウェアで、その間にファームウェアがあります。NISTの用語集は、変更の仕方で境界を引いています。ソフトウェアを「実行中に動的な書き込みや変更ができるコンピュータープログラムと関連データ」と定義し、ファームウェアは実行中には変更できないものとしています。ノートパソコンのオペレーティングシステムはソフトウェア、キーボードコントローラー内のコードはファームウェア、キーボードそのものはハードウェアです。

ソフトウェアの内部にも層があります。オペレーティングシステムやデバイスドライバーなどのシステムソフトウェアは、機械を管理します。アプリケーションソフトウェアは、表計算、予約サイト、メッセージングクライアントのように、人のために何かを行います。その間には、どちらを組み立てるときにも使う既製の部品であるライブラリやパッケージがあります。実際、現代の製品に含まれるコードの大半は、その製品のために書かれたものではなく、こうした形で取り込まれたものです。

もう一つ、後々多くを左右する区別があります。ソースコードは人が書き、読むもので、コンパイラーやインタープリターがそれをプロセッサーの実行する命令へ変換します。ソースコードも変換後のコードもソフトウェアですが、購入する際には同じものとして扱えません。コンパイル済みのファイルだけが納品されるなら、変更できるのは供給者だけです。

概念を判断に生かす

「ソフトウェアとは？」を実際の判断に生かしましょう。その概念を、開発または購入するシステムに結び付けて考えるお手伝いをします。 まず全体像を設計し、各部分のつながりと費用を、コードを書き始める前に明らかにします。

サービスを見る お問い合わせ →

用語の背後にある前提を検証する

解説とリンク先の実績を参考に、プロジェクトについての質問を整理してください。ツールを選ぶ前に、前提を一緒に検討できます。 14部門を支える共通インフラ計画 — 14 共通インフラ計画の対象となった研究部門

14部門を支える共通インフラ計画 お問い合わせ →

なぜ重要なのか

ソフトウェアを発注、購入、または借りる人は、その維持を誰が担うかも選んでいます。複製には費用がかからず、納品後の変更も機械の変更に比べれば安く済むという魅力は、ソフトウェアに終わりがない理由でもあります。構成部品には修正版が公開されます。オペレーティングシステムやブラウザーはインターフェースを廃止します。チームの誰も選んだ覚えのないライブラリで、欠陥が公表されます。その一つ一つが、コードを所有する側の仕事になります。

よくある選択肢では、その仕事の分担が異なります。

- **サービスとして借りる。** 更新とサーバーの運用は供給者が担います。利用者は、そのリリース日程、料金、乗り換えに伴う手間を引き受け、データは供給者側に置かれます。
- **ライセンスに基づいて自分で運用する。** 更新時期を自分で決められる一方、更新、バックアップ、セキュリティパッチの適用も自分で行います。
- **専用に開発してもらう。** 業務をソフトウェアに合わせるのではなく、ソフトウェアが業務に合います。ただし、稼働し続ける限り、その保守予算は自分で負担します。その選択が賢明だったかを決める費用は、公開後に生じます。
- **オープンソースを使う。** オープンソースの定義を満たすライセンスのもとで、利用、調査、変更、再配布ができます。それでも、更新を適用する人は必要です。入手費がかからないことと、運用費がかからないことは別です。

どの方法を選ぶ場合でも、契約する前に成果物について二つのことを確認する価値があります。最後に受け取れるのは何でしょうか。ソース、ビルド手順、内部の構成要素の一覧は含まれるでしょうか。そして、供給者はどうやって安全性を維持するのでしょうか。NISTのSecure Software Development Frameworkは、後者の対話も目的の一つとしており、開発の実践について購入者と供給者が共通の言葉で話せるようにします。前者については、NTIAの報告書がソフトウェア部品表を「ソフトウェアの構築に使われる各種構成要素の詳細と、サプライチェーン

上の関係を含む正式な記録」と定義しています。提出を求めれば、購入するソフトウェアのうち、他者が書いた部分がどれほどあるかを、少ない費用で確かめられます。

実際の活用例

何を出荷したのかを把握することは、チームが省きがちな部分です。あるプラットフォームでは、セキュリティを意見で判断する状態から、ビルドで確かめる状態へ移しました。継続的なセキュリティ施策では、AnchoreとSyftでビルドごとにソフトウェア部品表を生成し、Dependabotが6つのエコシステムを監視し、CodeQLとZAPがコードと稼働中のインスタンスを分析します。ビルド内で外部に依存する手順も固定し、サードパーティーのワークフローActionはすべてコミットハッシュで参照します。これはタグのように指す先を変えられません。自動化と並んで、脅威モデルを含む21件のセキュリティ監査が文書化されています。設定した検査が認識した脆弱性によって、ビルドを失敗させられます。代償は検出結果の量であり、終わりのないトリアージが必要です。

もう半分は、変更したソフトウェアを本番環境まで届けることです。誰かの記憶に頼ってリリースを行い、手作業の進め方も毎回少しずつ違っていた現場では、ビルド、テスト、デプロイの自動化によって、推計で4,000時間を削減し、リリースの速度と信頼性も向上しました。効果を生んだのは、誰かが一層注意深くなったことではなく、毎回の手順が同じになったことです。

ソフトウェアとハードウェアの境界も、修理台の上では実務上の問題になります。ノートパソコンの修理では、画面、ヒンジ、キーボード、マザーボードに加え、Linux、Windows、Macのソフトウェアの問題も扱いました。電源の入らないノートパソコンでは、原因は充電器かもしれず、基板やバッテリーかもしれません。そのため、何かに手を加える前に、実際の故障箇所を見つけることから始めました。

問題が起きるところ

脆弱性の公表が、1週間がかりの掘り起こし調査につながる

広く使われている構成要素で脆弱性が公表されても、組織内のどのシステムにそれが含まれるのか、誰も答えられません。構成要素の一覧は人の頭の中と、誰も読まないロックファイルにしかなく、時間に追われながら、システムごとに一覧を作り直さなければなりません。ビルドのたびに部品表を生成し、それが示すリリースと一緒に保管してください。そうすれば、「自分たちも影響を受けるか」という問いは検索で答えられます。

担当者によってリリースの成否が変わる

ある人がデプロイすると動き、別の人が行うと壊れます。違いを調べると、誰かの記憶の中にしかない手順が見つかります。毎回のリリースが小さな即興作業になっています。ビルド、テスト、デプロイを毎回同じ方法で行う自動ジョブに、一連の手順をすべて組み込み、残った手作業は取り除くべき欠陥として扱ってください。

公開すると予算が途切れる

2、3年が過ぎると、どの更新も危険に感じられ、フレームワークは複数のメジャーバージョン分だけ古くなり、提案されるのは書き直しだけになります。ソフトウェアにも建物と同じように一度費用を払い、その後は放置していた一方で、周囲は変わり続けていたのです。保守のための継続的な予算枠を確保し、変更がまだ小さいうちに依存関係を定期的に少しずつ更新し、土台となるプラットフォームも独自の日程で変わることを見込んでください。

名前は変わらないのに、指すコードが変わる

間にコミットがないのに、同じビルドが別の日には異なる結果を出します。どこかの参照先が、バージョン範囲、`latest` タグ、所有者が指す先を変えられるタグで参照したCIのActionなど、内容が変わり得るものになっています。サードパーティーのコードは正確なバージョンに固定し、タグの指す先を変更できる場合はコミットハッシュに固定してください。ロックファイルをコミットし、更新はボットに提案させれば、すべての変更をレビュー可能な差分として受け取れます。

参考資料

- NIST CSRC用語集：software — NISTの刊行物で使われる定義と、ソフトウェアをファームウェアから区別する定義です。
- NIST SP 800-218、Secure Software Development Framework — 脆弱性の少ないソフトウェアを構築するための基本的な実践を、購入者が供給者との間で使えるように記述しています。
- NTIA：ソフトウェア部品表の最小要素 — ソフトウェア内の構成要素の一覧を有用なものにするために、何を記録する必要があるかを説明しています。
- オープンソースの定義 — 公開されたソフトウェアがオープンソースに該当するために、ライセンスが満たすべき10の条件です。

用語集

実際の制約に合わせて選ぶ

業務の流れ、既存のインターフェース、変更を必要とする利用者をお知らせください。検証可能な納品内容と、お客様のチームが保守を続けられる引き継ぎ内容を定められます。

料金を見る お問い合わせ →

用語を実務に結び付ける

「ソフトウェアとは？」がお客様の組織にどう当てはまるかをご相談ください。現在のシステムの具体例が一つあれば、話を始められます。

サービスを見る

welcome@engineer.company +45 50 10 61 84 デンマーク、コペンハーゲン

このページを持ち歩く

連絡先カード

心を込めて丁寧に制作： D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

著作権 © 2025 – 現在 · Engineer ApS · すべての権利を留保します。

印刷元： https://engineer.company/ja/glossary/software/