Engineer

Програмне забезпечення

програми й дані, з якими працює комп'ютер

Що це таке

Програмне забезпечення — це все в комп’ютерній системі, що змінюють, переписуючи інструкції, а не перепаюючи плату: програми, дані, які вони несуть із собою, і налаштування, що керують їхньою поведінкою. У документах його скорочують до ПЗ, у розмові кажуть «софт», окрему одиницю називають програмою, а те, що людина відкриває, щоб виконати якусь роботу, зазвичай зветься застосунком. Словом «код» найчастіше позначають вихідний текст, який редагує розробник.

Протилежне поняття — апаратне забезпечення, тобто сама машина, а між ними лежить вбудоване програмне забезпечення, або прошивка (firmware). Глосарій NIST проводить межу через можливість змін: програмне забезпечення — це програми й пов’язані з ними дані, які можна записувати чи змінювати під час виконання, а прошивку під час роботи змінити не можна. Операційна система ноутбука — це ПЗ, код у контролері клавіатури — прошивка, а сама клавіатура — залізо.

Усередині ПЗ є шари. Системне програмне забезпечення, як-от операційна система чи драйвер, керує машиною. Прикладне робить щось для людини: електронна таблиця, сайт бронювання, клієнт для повідомлень. Між ними лежать бібліотеки й пакети, готові частини, з яких складають і те, і інше. У сучасному продукті більша частина коду потрапила саме так, а не була написана спеціально для нього.

Ще одна відмінність стає вирішальною пізніше. Вихідний код пишуть і читають люди, а компілятор або інтерпретатор перетворює його на інструкції, які виконує процесор. І те, і те — програмне забезпечення, але під час купівлі вони не взаємозамінні. Якщо замовник отримує лише скомпільовані файли, змінювати їх зможе тільки постачальник.

Застосуйте поняття до рішення

Застосуйте «Що таке програмне забезпечення?» до конкретного рішення. Ми можемо пов'язати поняття із системою, яку ви створюєте або купуєте. Структура системи, спершу продумана на папері, щоб частини пасували одна до одної, а вартість була відома до першого рядка коду.

Переглянути наші послуги Зв'язатися з нами →

Перевірте припущення за терміном

Використайте пояснення та пов'язані роботи, щоб сформулювати запитання для проєкту. Ми можемо перевірити припущення разом до вибору інструмента. Один інфраструктурний план для 14 підрозділів — 14 дослідницьких підрозділів, охоплених спільним інфраструктурним планом

Один інфраструктурний план для 14 підрозділів Зв'язатися з нами →

Чому це важливо

Той, хто замовляє, купує чи орендує програмне забезпечення, водночас обирає, хто його супроводжуватиме. Те, що робить ПЗ привабливим, — копія нічого не коштує, а зміна після постачання дешева порівняно з переробкою машини, — пояснює й те, чому воно ніколи не буває завершеним. Компоненти отримують виправлення. Операційні системи й браузери відмовляються від старих інтерфейсів. Одного дня оприлюднюють вразливість у бібліотеці, яку ніхто в команді не пригадує, щоб обирав. Усе це стає роботою для власника коду.

Поширені шляхи розподіляють цю роботу по-різному:

- **Оренда як сервісу.** Оновлення й сервери лишаються на боці постачальника. Натомість ви успадковуєте його графік випусків, ціни й клопіт із переходом деінде, а дані зберігаються в нього.
- **Ліцензія і власна експлуатація.** Коли оновлюватися, вирішуєте ви, але й оновлення, резервні копії та латки безпеки теж ваші.
- **Розробка на замовлення.** ПЗ підлаштовується під процес, а не навпаки, проте супровід лишається на власникові, доки система працює. Чи було рішення розумним, покажуть витрати після запуску.
- **Відкритий код.** Ліцензія, що відповідає Open Source Definition, дозволяє використовувати, вивчати, змінювати й поширювати код. Оновлення однаково хтось має встановлювати: безкоштовно отримати і безкоштовно експлуатувати — різні речі.

Хоч би який шлях ви обрали, перед підписанням варто поставити два питання. Що ви отримаєте наприкінці: вихідний код, інструкції зі збирання і перелік того, з чого ПЗ складається? І як постачальник підтримуватиме його безпеку? Secure Software Development Framework від NIST написано зокрема для другої розмови: він дає замовникові й постачальникові спільну мову про

практики розробки. Щодо першого питання, звіт NTIA описує software bill of materials, скорочено SBOM, як формальний запис про компоненти, з яких зібрано ПЗ, і про їхні зв'язки в ланцюгу постачання. Попросити такий перелік — дешевий спосіб побачити, яку частину оплаченого написали інші люди.

Як це виглядає на практиці

На одній платформі безпеку перенесли з думок у сам процес збирання. У межах безперервної програми безпеки SBOM формується під час кожного збирання за допомогою Anchore і Syft, Dependabot стежить за 6 екосистемами, CodeQL аналізує код, ZAP перевіряє запущений екземпляр, а кожну сторонню GitHub Action закріплено за хешем коміту, а не за тегом. Поряд з автоматизацією є 21 письмовий аудит безпеки, серед них модель загроз. Вразливість, яку розпізнають налаштовані перевірки, може зупинити збірку. Сторінка чесно називає й ціну: знахідок багато, і їх доводиться сортувати без кінця.

Шлях від зміни до робочого середовища — друга половина справи. Там, де випуски трималися на тому, що хтось пам'ятав послідовність ручних кроків і щоразу виконував її трохи інакше, автоматизовані конвеєри збирання, тестування й розгортання заощадили, за оцінкою, 4 000 годин, а випуски стали швидшими й надійнішими. Виграш дала однаковість процесу, а не додаткова обережність людей.

Межа між програмами й залізом дуже відчутна на ремонтному столі. Ремонт ноутбуків охоплював екрани, шарніри, клавіатури й материнські плати, а також програмні проблеми в Linux, Windows і Mac. Ноутбук, який не вмикається, може мати несправний зарядний пристрій, плату чи батарею, тому робота починалася з пошуку справжньої причини, ще до того, як щось розбирати.

Де виникають проблеми

Попередження про вразливість обертається тижнем розкопок

Оголошують вразливість у поширеному компоненті, і ніхто не може сказати, які системи організації його містять. Перелік існував лише в головах і в lock-файлах, яких ніхто не читає, тож його доводиться відтворювати під тиском,

систему за системою. Формуйте SBOM під час кожного збирання і зберігайте його поряд із випуском, який він описує. Тоді питання «чи нас це зачіпає?» стає звичайним пошуком.

Випуск залежить від того, хто його робить

Розгортання працює, коли його запускає одна людина, і ламається в руках іншої, а різниця виявляється кроком, що існує лише в чиїйсь пам'яті. Кожен випуск перетворюється на маленьку імпровізацію. Перенесіть увесь шлях в автоматизований процес CI/CD, який щоразу однаково збирає, тестує й розгортає, а кожен ручний крок, що лишився, вважайте дефектом, який треба прибрати.

Бюджет закінчується в день запуску

Через два-три роки кожне оновлення здається ризикованим, фреймворк відстає на кілька мажорних версій, і єдина пропозиція на столі — переписати все. ПЗ оплатили як будівлю й залишили в спокої, а все довкола рухалося далі. Закладайте постійну статтю на супровід, оновлюйте залежності малими регулярними кроками, поки кожен крок ще малий, і зважайте на те, що платформи під вами змінюються у власному темпі.

Назва та сама, а код за нею інший

Те саме збирання дає різні результати два дні поспіль без жодного коміту між ними. Десь посилання вказує на рухому ціль: діапазон версій, тег `latest` або дію CI, підключену через тег, який її власник може перенести. Закріплюйте сторонній код за точною версією, а там, де теги рухаються, — за хешем коміту, тримайте lock-файл у репозиторії й доручіть боту пропонувати оновлення, щоб кожна зміна надходила як diff, який можна переглянути.

Що почитати далі

- NIST CSRC Glossary: software — визначення, якими користуються публікації NIST, зокрема те, що відокремлює ПЗ від прошивки.
- NIST SP 800-218, Secure Software Development Framework — базові практики розробки ПЗ з меншою кількістю вразливостей, на які замовник може спиратися в розмові з постачальником.
- NTIA: The Minimum Elements for a Software Bill of Materials — що має містити перелік компонентів ПЗ, щоб від нього була користь.
- The Open Source Definition — десять критеріїв, яким має відповідати ліцензія, щоб ПЗ під нею вважалося відкритим.

Словник

Обирайте за фактичними умовами

Опишіть робочий процес, наявні інтерфейси та користувачів, яким потрібна зміна. Ми можемо визначити перевірюваний результат і спосіб передачі, який дасть вашій команді змогу підтримувати його.

Дивитися, скільки це коштує Зв'язатися з нами →

Зробіть термінологію практичною

Запитайте, як «Що таке програмне забезпечення?» стосується вашої організації. Конкретного прикладу з поточної системи достатньо для початку розмови.

Переглянути наші послуги

welcome@engineer.company +45 50 10 61 84 Копенгаген, Данія

Візьміть цю сторінку із собою

Візитівка

Дбайливо зроблено з любов'ю — D.

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

Авторське право © 2025 – дотепер · Engineer ApS · Усі права захищено.

Надруковано з https://engineer.company/uk/glossary/software/