Engineer

logiciel

Programmes et données utilisés par un ordinateur

Ce que c’est

Le logiciel désigne toute partie d’un système informatique que l’on peut modifier en écrivant plutôt qu’en soudant : les programmes, les données qu’ils transportent et les paramètres qui leur indiquent comment se comporter. En anglais, software est un nom indénombrable : un élément se dit donc « a piece of software » (un logiciel) ou « a program » (un programme). Ce qu’une personne ouvre pour accomplir une tâche est généralement appelé une application. « Code » désigne plutôt le texte source que le développeur modifie, et les spécifications abrègent parfois l’ensemble en SW.

On l’oppose habituellement au matériel, la machine physique, avec le micrologiciel entre les deux. Le glossaire du NIST trace cette frontière selon la possibilité de modification : le logiciel comprend les « programmes informatiques et données associées qui peuvent être écrits ou modifiés dynamiquement pendant l’exécution », tandis que le micrologiciel est celui qui ne peut pas être modifié pendant son exécution. Le système d’exploitation d’un ordinateur portable est un logiciel ; le code du contrôleur de son clavier est un micrologiciel ; le clavier est du matériel.

Le logiciel comporte plusieurs couches. Les logiciels système, comme un système d’exploitation ou un pilote de périphérique, gèrent la machine. Les logiciels applicatifs accomplissent quelque chose pour une personne : un tableur, un site de réservation, un client de messagerie. Entre les deux se trouvent les bibliothèques et les paquets, des éléments prêts à l’emploi à partir desquels les deux catégories sont assemblées. En pratique, la majeure partie du code d’un produit moderne est arrivée ainsi, plutôt que d’avoir été écrite pour lui.

Une autre distinction détermine beaucoup de choses par la suite. Le code source est ce que les personnes écrivent et lisent ; un compilateur ou un interpréteur le transforme en instructions exécutées par un processeur. Ces deux formes sont du logiciel, mais ne sont pas interchangeables à l'achat. Une livraison limitée à des fichiers compilés ne peut être modifiée que par son fournisseur.

Utilisez le concept pour prendre une décision

Utilisez « Qu'est-ce qu'un logiciel ? » pour éclairer une décision réelle. Nous pouvons vous aider à relier ce concept au système que vous construisez ou achetez. Définir d'abord le système sur le papier pour que ses éléments s'articulent et que son coût soit connu avant d'écrire du code.

Découvrir nos prestations Contactez-nous →

Vérifiez les hypothèses derrière le terme

Appuyez-vous sur l'explication et les réalisations liées pour formuler les questions de votre projet. Nous pouvons examiner les hypothèses avec vous avant le choix d'un outil. Un plan d'infrastructure pour 14 départements — 14 départements de recherche sur un même plan d'infrastructure

Un plan d'infrastructure pour 14 départements Contactez-nous →

Pourquoi c'est important

Faire réaliser, acheter ou louer un logiciel revient à choisir qui en assumera l'entretien. Ce qui le rend attrayant — le copier ne coûte rien et le modifier après livraison coûte peu par rapport à la modification d'une machine — explique aussi pourquoi il n'est jamais terminé. Les composants qui le constituent publient des correctifs. Les systèmes d'exploitation et les navigateurs retirent des interfaces. Une faille est divulguée dans une bibliothèque que personne dans l'équipe ne se souvient avoir choisie. Chacun de ces événements crée du travail pour le responsable du code.

Les formules courantes répartissent ce travail différemment :

- **Loué sous forme de service.** Le fournisseur gère les mises à jour et les serveurs. Vous héritez de son calendrier de versions, de ses tarifs et de l'effort nécessaire pour partir, et vos données résident chez lui.
- **Sous licence et exploité par vos soins.** Vous choisissez quand effectuer les mises à niveau, et vous assurez aussi ces mises à niveau, les sauvegardes et les correctifs de sécurité.
- **Construit pour vous.** Il s'adapte au processus plutôt que l'inverse, et son budget de maintenance vous incombe aussi longtemps qu'il fonctionne. Le coût qui détermine si ce choix était judicieux vient après le lancement.
- **Open source.** Une licence conforme à l'Open Source Definition permet de l'utiliser, de l'étudier, de le modifier et de le redistribuer. Quelqu'un doit tout de même appliquer les mises à jour : l'obtenir gratuitement et l'exploiter gratuitement sont deux propriétés différentes.

Quelle que soit la formule, deux questions méritent d'être posées avant de signer. Que recevez-vous à la fin : le code source, les instructions pour produire le logiciel et la liste de ses composants ? Et comment le fournisseur maintiendra-t-il sa sécurité ? Le Secure Software Development Framework du NIST existe notamment pour cette seconde discussion : il donne aux acheteurs et aux fournisseurs un vocabulaire commun sur les pratiques de développement. Pour la première, un rapport de la NTIA définit une nomenclature logicielle comme « un relevé formel contenant les détails et les relations de chaîne d'approvisionnement des différents composants utilisés pour construire un logiciel ». En demander une coûte peu et permet de voir quelle part de ce que vous payez a été écrite par d'autres.

À quoi cela ressemble en pratique

Savoir ce qui a été livré est l'étape que les équipes omettent. Sur une plateforme, la sécurité est passée d'une opinion à un résultat de construction : dans un programme continu de sécurité, une nomenclature logicielle est produite à chaque construction avec Anchore et Syft. Dependabot surveille 6 écosystèmes, CodeQL et ZAP analysent le code et l'instance en cours d'exécution. Même les étapes externes de la construction sont figées : chaque Action tierce du workflow est référencée par son empreinte de commit, que personne ne peut déplacer comme on déplacerait une étiquette. À côté de l'automatisation figurent 21 audits de sécurité écrits, dont un modèle de menaces. Une vulnérabilité reconnue par les contrôles configurés peut faire échouer la construction. Le prix à payer est le volume des constats, qui exige un triage permanent.

L'autre moitié consiste à faire passer une modification jusqu'à la production. Là où les versions dépendaient d'une personne se souvenant d'une suite d'étapes manuelles et les exécutant chaque fois un peu différemment, l'automatisation de la construction, des tests et du déploiement a permis d'économiser un total estimé à 4 000 heures et a rendu les mises en production plus rapides et plus fiables. Le gain venait d'un parcours identique à chaque fois, plutôt que d'une plus grande attention de chacun.

Enfin, la frontière entre logiciel et matériel est concrète sur un établi de réparation. Les réparations d'ordinateurs portables couvraient écrans, charnières, claviers et cartes mères, ainsi que les problèmes logiciels sous Linux, Windows et Mac. Un portable qui ne s'allume pas peut avoir un chargeur, une carte ou une batterie défectueux : le travail commençait donc par la recherche de la véritable panne avant toute intervention.

Où les problèmes surviennent

Un avis de vulnérabilité devient une semaine d'archéologie

Une vulnérabilité est annoncée dans un composant très répandu, et personne ne sait quels systèmes de l'organisation le contiennent. L'inventaire n'a jamais existé que dans les têtes et dans des fichiers de verrouillage que personne ne lit. Il faut donc le reconstituer dans l'urgence, système par système. Produisez une nomenclature à chaque construction et conservez-la avec la version qu'elle décrit. La question « sommes-nous concernés ? » devient alors une recherche.

Les mises en production dépendent de la personne qui les exécute

Le déploiement fonctionne pour une personne et échoue pour une autre. La différence se révèle être une étape conservée dans la mémoire de quelqu'un. Chaque mise en production est une petite improvisation. Placez tout le parcours dans une tâche automatisée qui construit, teste et déploie de la même façon à chaque fois, et traitez toute étape manuelle restante comme un défaut à supprimer.

Le budget s'arrête au lancement

Deux ou trois ans plus tard, chaque mise à jour semble risquée, le framework a plusieurs versions majeures de retard et la seule proposition est une réécriture. Le logiciel a été payé comme un bâtiment, puis abandonné à lui-même pendant que tout changeait autour. Conservez une enveloppe permanente pour la maintenance, mettez les dépendances à niveau régulièrement par petites étapes tant qu'elles sont encore petites, et attendez-vous à ce que les plateformes sous-jacentes évoluent selon leur propre calendrier.

Le nom ne change pas, mais le code qu'il désigne change

Une même construction produit des résultats différents à deux dates, sans aucun commit entre les deux. Quelque part, une référence pointe vers une cible mouvante : une plage de versions, une étiquette `latest`, une action de CI référencée par une étiquette que son propriétaire peut déplacer. Fixez le code tiers à une version exacte ou, lorsque les étiquettes peuvent bouger, à une empreinte de commit. Versionnez le fichier de verrouillage et confiez les propositions de mise à jour à un robot : chaque changement arrive ainsi sous forme de différences examinables.

Pour aller plus loin

- Glossaire NIST CSRC : logiciel — les définitions utilisées par les publications du NIST, dont celle qui distingue logiciel et micrologiciel.
- NIST SP 800-218, Secure Software Development Framework — un socle de pratiques pour construire des logiciels comportant moins de vulnérabilités, rédigé pour que les acheteurs puissent l'utiliser avec leurs fournisseurs.
- NTIA : les éléments minimaux d'une nomenclature logicielle — les informations qu'une liste des composants d'un logiciel doit consigner pour être utile.
- Open Source Definition — les dix critères qu'une licence doit respecter pour que le logiciel publié sous celle-ci soit considéré comme open source.

Glossaire

Choisissez selon vos contraintes réelles

Présentez le processus de travail, les interfaces existantes et les utilisateurs qui ont besoin du changement. Nous pouvons définir une livraison testable et une transmission permettant à votre équipe d'assurer la maintenance des éléments livrés.

Consulter les tarifs Contactez-nous →

Rendez la terminologie concrète

Demandez comment « Qu'est-ce qu'un logiciel ? » s'applique à votre organisation. Un exemple concret tiré de votre système actuel suffit pour engager la discussion.

Découvrir nos prestations

welcome@engineer.company +45 50 10 61 84 Copenhague, Danemark

Emportez cette page avec vous

Fiche de contact

Créé avec soin et amour par D.

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

Droit d'auteur © 2025 – aujourd'hui · Engineer ApS · Tous droits réservés.

Imprimé depuis https://engineer.company/fr/glossary/software/