Engineer

software

Programas y datos que utiliza un ordenador

Qué es

El software comprende las partes de un sistema informático que se pueden cambiar escribiendo en lugar de soldando: los programas, los datos que contienen y los ajustes que determinan su comportamiento. En inglés, la palabra es un sustantivo incontable. Por eso una unidad se llama «a piece of software» («una pieza de software») o «a program» («un programa»), y lo que una persona abre para hacer una tarea suele llamarse aplicación. «Código» suele referirse al texto fuente que edita quien desarrolla, y las especificaciones a veces abrevian el conjunto como SW.

Su opuesto habitual es el hardware, la máquina física, con el firmware entre ambos. El glosario del NIST traza la frontera según la posibilidad de cambio: el software se define como «programas informáticos y datos asociados que pueden escribirse o modificarse dinámicamente durante la ejecución», mientras que el firmware no puede modificarse mientras se ejecuta. El sistema operativo de un portátil es software; el código del controlador de su teclado es firmware; el teclado es hardware.

Dentro del software hay capas. El software de sistema, como un sistema operativo o un controlador de dispositivo, gestiona la máquina. El software de aplicación hace algo para una persona: una hoja de cálculo, un sitio de reservas o un cliente de mensajería. Entre ambos están las bibliotecas y los paquetes, piezas ya hechas con las que se ensamblan las dos clases. En la práctica, la mayor parte del código de un producto moderno llegó de ese modo, en lugar de escribirse para él.

Otra distinción condiciona muchas decisiones posteriores. El código fuente es lo que las personas escriben y leen; un compilador o un intérprete lo convierte en instrucciones que ejecuta un procesador. Tanto el código fuente como el código

compilado son software, pero no son intercambiables al comprarlos. Si solo se entregan archivos compilados, únicamente el proveedor puede modificarlos.

Utilice el concepto en una decisión

Aplique «¿Qué es el software?» a una decisión real. Podemos ayudarle a relacionar el concepto con el sistema que está construyendo o comprando. La estructura del sistema se define primero sobre el papel para que las piezas encajen y los costes se conozcan antes de escribir código.

Explore nuestros servicios Póngase en contacto →

Compruebe los supuestos detrás del término

Utilice la explicación y los trabajos enlazados para formular preguntas sobre su proyecto. Podemos examinar los supuestos con usted antes de elegir una herramienta. Un plan de infraestructura para 14 departamentos — 14 departamentos de investigación con un único plan de infraestructura

Un plan de infraestructura para 14 departamentos Póngase en contacto →

Por qué importa

Quien encarga, compra o alquila software está eligiendo quién se hará cargo de mantenerlo. Lo que lo hace atractivo, que copiarlo no cueste nada y que cambiarlo después de la entrega resulte barato frente a cambiar una máquina, también explica por qué nunca está terminado. Los componentes con los que se construyó publican correcciones. Los sistemas operativos y los navegadores retiran interfaces. Se divulga una vulnerabilidad en una biblioteca que nadie del equipo recuerda haber elegido. Todo ello se convierte en trabajo para quien tenga el código a su cargo.

Las opciones habituales reparten ese trabajo de distintas maneras:

- **Alquilado como servicio.** El proveedor se ocupa de las actualizaciones y los servidores. Usted asume su calendario de publicaciones, sus precios y el esfuerzo de abandonarlo, y los datos quedan en sus sistemas.
- **Con licencia y operado por usted.** Usted elige cuándo actualizar. También realiza las actualizaciones, las copias de seguridad y la aplicación de parches de seguridad.
- **Hecho a medida.** Se adapta al proceso, en lugar de exigir lo contrario, y el presupuesto de mantenimiento le corresponde mientras siga funcionando. El coste que decide si fue una buena elección llega después del lanzamiento.
- **De código abierto.** Una licencia que cumpla la definición de código abierto permite usarlo, estudiarlo, modificarlo y redistribuirlo. Alguien debe seguir aplicando las actualizaciones: obtenerlo gratis y operarlo gratis son propiedades distintas.

Sea cual sea la opción, conviene plantear dos preguntas sobre el resultado antes de firmar. ¿Qué se recibe al final: el código fuente, las instrucciones para generar una versión a partir de él y una lista de lo que contiene? ¿Y cómo mantendrá su seguridad el proveedor? El marco de desarrollo de software seguro del NIST existe en parte para esa segunda conversación: ofrece a compradores y proveedores un vocabulario común sobre las prácticas de desarrollo. Para la primera, un informe de la NTIA define una lista de materiales de software como «un registro formal que contiene los detalles y las relaciones de la cadena de suministro de los distintos componentes utilizados para construir software». Pedir una cuesta poco y permite ver cuánto de lo que se está pagando fue escrito por otras personas.

Cómo se aplica en la práctica

Saber qué se entregó es la parte que los equipos omiten. En una plataforma, el trabajo de seguridad pasó de ser una opinión a formar parte de la compilación: en un programa continuo de seguridad se genera una lista de materiales de software en cada compilación con Anchore y Syft. Dependabot vigila 6 ecosistemas, y CodeQL y ZAP analizan el código y la instancia en ejecución. Incluso los pasos externos de la compilación están fijados: cada Action de terceros de los flujos de trabajo se referencia por su hash de commit, que nadie puede mover como se mueve una etiqueta. Junto a la automatización hay 21 auditorías de seguridad escritas, incluido un modelo de amenazas. Una vulnerabilidad reconocida por las comprobaciones configuradas puede hacer fallar la compilación. El coste es el volumen de hallazgos, que exige una evaluación y priorización continuas.

Llevar el software desde un cambio hasta producción es la otra mitad. Donde las publicaciones dependían de que alguien recordara una secuencia de pasos manuales y la realizara de forma algo distinta cada vez, automatizar la compilación, las pruebas y el despliegue ahorró unas 4.000 horas según la estimación y volvió las publicaciones más rápidas y fiables. La mejora procedía de seguir siempre el mismo recorrido, no de que alguien tuviera más cuidado.

Y la frontera entre software y hardware es práctica en un banco de reparación. Las reparaciones de portátiles abarcaron pantallas, bisagras, teclados y placas base, además de problemas de software en Linux, Windows y Mac. Un portátil que no enciende puede tener un problema en el cargador, la placa o la batería. Por eso el trabajo empezaba por localizar el fallo real antes de tocar nada.

Dónde surgen los problemas

Un aviso de seguridad se convierte en una semana de arqueología

Se anuncia una vulnerabilidad en un componente muy utilizado y nadie puede decir qué sistemas de la organización lo contienen. El inventario solo existía en la memoria de las personas y en archivos de bloqueo que nadie lee. Hay que reconstruirlo bajo presión, sistema por sistema. Genere una lista de materiales en cada compilación y consérvela junto a la versión que describe. La pregunta «¿nos afecta?» se convierte así en una búsqueda.

Las publicaciones dependen de quién las haga

El despliegue funciona cuando lo ejecuta una persona y falla cuando lo hace otra. La diferencia resulta ser un paso que solo existe en la memoria de alguien. Cada publicación es una pequeña improvisación. Incluya el recorrido completo en una tarea automatizada que compile, pruebe y despliegue siempre del mismo modo, y trate cualquier paso manual que quede como un defecto que debe eliminarse.

El presupuesto termina en el lanzamiento

Dos o tres años después, cada actualización parece arriesgada, el framework lleva varias versiones principales de retraso y la única propuesta es reescribirlo. El software se pagó como si fuera un edificio y después se dejó intacto, mientras todo a su alrededor seguía cambiando. Reserve un presupuesto permanente para mantenimiento, actualice las dependencias en pequeños pasos regulares mientras cada paso aún sea pequeño y cuente con que las plataformas subyacentes cambiarán según sus propios calendarios.

El nombre permanece mientras cambia el código al que apunta

La misma compilación produce resultados distintos en dos días sin ningún commit entre ellos. En algún lugar, una referencia apunta a algo que cambia: un intervalo de versiones, una etiqueta `latest` o una acción de CI referenciada mediante una etiqueta que su propietario puede redirigir. Fije el código de terceros a una versión exacta o, cuando las etiquetas puedan moverse, a un hash de commit. Registre el archivo de bloqueo en el control de versiones y deje que un bot proponga las actualizaciones, para que cada cambio llegue como un diff revisable.

Lecturas adicionales

- Glosario del NIST CSRC: software — Las definiciones que utilizan las publicaciones del NIST, incluida la que distingue software y firmware.
- NIST SP 800-218, marco de desarrollo de software seguro — Un conjunto básico de prácticas para construir software con menos vulnerabilidades, redactado para que los compradores puedan utilizarlo con sus proveedores.
- NTIA: elementos mínimos de una lista de materiales de software — Lo que debe registrar una lista de los componentes de un programa para resultar útil.
- Definición de código abierto — Los diez criterios que debe cumplir una licencia para que el software publicado con ella se considere de código abierto.

Glosario

Elija según sus restricciones reales

Describa el flujo de trabajo, las interfaces existentes y los usuarios que necesitan el cambio. Podemos definir una entrega comprobable y un traspaso que permita a su equipo mantener lo entregado.

Vea cuánto cuesta Póngase en contacto →

Haga práctica la terminología

Pregunte cómo se aplica «¿Qué es el software?» a su organización. Basta un ejemplo concreto de su sistema actual para iniciar la conversación.

Explore nuestros servicios

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

Llévese esta página

Tarjeta de contacto

Creado con esmero y cariño por D.

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

Derechos de autor © 2025 – actualidad · Engineer ApS · Todos los derechos reservados.

Impreso desde https://engineer.company/es/glossary/software/