

OBSOLESCENCIA DE SOFTWARE

Grupo Sem-ES209

ARQUITECTURA DE SOFTWARE

Iyán Fernández Riol – UO288231

CURSO 2023-24

Carmen Espinosa Martínez – UO289659

María López García-Consuegra – UO289029

¿Qué es la obsolescencia de software?

La obsolescencia del software es un proceso en el que un software se vuelve desactualizado o no funcional.

Este fenómeno está en constante crecimiento, de la mano con el avance en la tecnología. Con la rápida innovación, las versiones anteriores pueden volverse obsoletas rápidamente, impulsando la necesidad de actualizaciones constantes para mantener la seguridad y el rendimiento. El software desactualizado puede presentar problemas de compatibilidad y vulnerabilidades de seguridad.

Tipos de obsolescencia del software

La obsolescencia del software puede ocurrir en 3 áreas diferentes:

- **Habilidades:** son las habilidades necesarias para desarrollar, mantener y modificar software. La pérdida de dichas habilidades se considera obsolescencia. Un ejemplo, es la dificultad de mantener el software heredado, es decir, utilizando lenguajes de programación heredados. Estos son lenguajes que han sido ampliamente usados en el pasado, pero sustituidos por otros más modernos.
- **Software comercial:** el software COTS o software comercial se vuelve obsoleto cuando su proveedor deja de ofrecer soporte. Este es el problema de obsolescencia de software más común y arriesgado, pues es difícil predecir cuándo va a suceder. El ejemplo más típico de obsolescencia de software comercial es el sistema operativo Windows XP, lanzado en 2001 y ampliamente utilizado durante muchos años. Sin embargo, en 2014 dejó de ofrecer soporte técnico.
- **Medios de almacenamiento:** representa los materiales y formatos de almacenamiento de datos utilizados para mantener la información del software. En este caso, existe el riesgo de perder datos e información. Un ejemplo de esta última obsolescencia puede ser el uso de disquetes como medios de almacenamiento de datos, sustituidos por otros medios más avanzados como los CD o DVD.

Causas

Existen diferentes causas que originan la obsolescencia de software, entre las que se encuentran:

- Dificultad para coordinar el mantenimiento del software entre diferentes organizaciones
- Complejidad progresiva en los sistemas tecnológicos
- La adición de nuevas funcionalidades implica potenciales nuevas vulnerabilidades.
- Obsolescencia programada (se verá más adelante)

Consecuencias

La obsolescencia de software tiene varias consecuencias que pueden resultar muy dañinas para los usuarios de ese software. Algunas de ellas son:

- Fallos en cascada: cuando se explota una vulnerabilidad de un componente obsoleto, se pone en peligro todo el sistema al que está conectado ese componente.
- Aumento de riesgo si hay interconexión de componentes: cuando se realiza un cambio en un dispositivo, se amplía el impacto de este a todos los dispositivos conectados a él.
- Puede poner en peligro la seguridad y estabilidad de sistemas críticos.

Por todas estas vulnerabilidades y otras que no hemos mencionado, es muy importante mantener el software siempre actualizado.

Vida útil y mantenimiento del software

Hay muchos factores que influyen en la vida útil y el mantenimiento del software y en el podcast se habla sobre varios de ellos a través de ejemplos. El factor que más se menciona es el modelo de negocio que lleve la empresa, por ejemplo, empresas con servicios en línea como puede ser Google, Facebook, etc, tienen muchas facilidades a la hora de mantener su producto ya que usan actualizaciones en línea desde sus servidores haciendo que el mantenimiento sea rápido y sin que el usuario se entere o deje de tener disponible el producto. Por otro lado, hay empresas como las dedicadas a dispositivos IoT que tienen mucho más difícil el mantenimiento y el alargamiento de la vida útil ya que sus dispositivos no se pueden actualizar en línea y el hardware puede acabar obsoleto rápidamente. Otro factor que se menciona es la dedicación al mantenimiento e intención de la empresa, algunas de ellas no creen rentable tener que estar manteniendo su software durante largos periodos de tiempo ya que quieren que el usuario utilice un modelo más nuevo de su

producto, en vez de utilizar uno que tiene 5 años de antigüedad, por ejemplo, el entrevistado dice que suele pasar con las empresas de coches y su software GPS.

Anderson comenta que hay empresas que, cuando se encuentran vulnerabilidades que deberían arreglar para alargar la vida útil y la seguridad de su producto, deciden ignorar o decir que eso no es su problema, haciendo así que no se mantenga el software y pueda tener fallos de seguridad importantes. Como último factor está el coste del mantenimiento, Anderson dice que en la empresa Tata Consulting, tienen 90 días de mantenimiento y que podría ser buena idea para aumentar el mantenimiento poder ofrecer mantenimiento a largo plazo, obviamente habría que pagar por ello. También se comenta que, si se cambia el mantenimiento ofrecido por los buenos ingenieros de una empresa por otros de coste más barato puede hacer que, aunque el mantenimiento esté presente, no sirva para nada e incluso haga que haya más problemas de seguridad con más frecuencia. Finalmente, el entrevistado habla de que el coste de revisar y actualizar ciertos sectores que usan software antiguo es demasiado alto, por ejemplo, en ciertas estaciones nucleares se sigue usando Windows 95, pero no se cambia porque el coste es demasiado alto.

Prevención

¿Cómo se previene la obsolescencia del software? No siempre es sencillo, pero podría resultar beneficioso aplicar medidas como mejorar la coordinación entre empresas y promover la educación sobre mantenibilidad y seguridad, tanto en empresas como en centros de formación. Es fundamental también priorizar la corrección de vulnerabilidades sobre los avances en funcionalidades, dado que priorizar los avances podría llevarnos a una acumulación seria de vulnerabilidades en nuestro sistema. Además, se debe reducir la dependencia de terceros en las soluciones tecnológicas para garantizar la continuidad operativa.

Diferencias entre modelos de negocio

Los diferentes modelos de negocio tienen grandes diferencias en cuanto a cómo tratan la obsolescencia. Google y Facebook son servicios en línea que pierden dinero si no arreglan sus fallos rápidamente, mientras que el software de los coches o trenes son productos físicos que también deben arreglar los fallos cuanto antes por posibles problemas de seguridad, pero es mucho más difícil que con los servicios en línea. Los productos del Internet of Things (IoT) son productos electrónicos que no tienen tanta urgencia en solventar los fallos tan rápido como las otras empresas.

Los servicios en línea tienen mucha facilidad para arreglar sus fallos porque ejecutan sus actualizaciones a través de sus servidores y, gracias a ello, mantienen los productos que tienen en todo el mundo de manera instantánea. Este mantenimiento es fácil y permite tenerlo a largo plazo ya que estos servicios suelen estar cambiando constantemente. Por otro lado, los productos físicos o electrónicos también deben tener mantenimiento a largo

plazo, aunque les es bastante más difícil ya que no pueden hacer actualizaciones en línea, sino que necesitan de asistencia presencial para realizar las actualizaciones pertinentes. Para los productos electrónicos este mantenimiento es casi inexistente, ya que si se distribuyen por todo el mundo y la empresa solo está en un solo país, es imposible ir uno a uno con cada producto distribuido por el mundo.

Obsolescencia programada

Es importante también, la mención a la obsolescencia programada. Esta se refiere a la programación del fin de la vida útil de un producto y tiene como objetivo el aumento de los ingresos por parte de los fabricantes. Este tipo de obsolescencia es ampliamente utilizada en productos del día a día. Además, puede verse en el software a través de lanzamientos de nuevas actualizaciones y versiones no compatibles con las anteriores.

En resumen, se trata de una estrategia comercial totalmente intencionada en la que los productos se vuelven obsoletos tras un periodo de tiempo fijado por fabricantes y diseñadores.

Bibliografía

Podcast 559 Ross Anderson en Software Obsolence. <https://se-radio.net/2023/04/se-radio-559-ross-anderson-on-software-obsolence/>

Documento sobre desafíos clave en la gestión de la obsolescencia del software para sistemas industriales de productos y servicios. <https://ep.liu.se/ecp/077/050/ecp10077050.pdf>