

# Software Obsolescence

Manuel González Santos – UO279387

## Ross Anderson

- Professor of Security Engineering at the [Department of Computer Science and Technology, University of Cambridge](#) where he was part of the University's security group.
- Falleció 28/03/2024

## Obsolescence

**Would you be able to explain what is obsolescence or end of software life?**

- Nuevas features => features interaccionan entre sí => dependability issues
- Software as EcoSystem donde el sistema base es el OS y cuando este recibe una actualización tendrá influencia en algún otro componente del ecosistema, pasa lo mismo entre grupos más pequeños
- Cada nodo del software tiene todo un ecosistema en el que vive

**when a software gets obsolete, does that mean nothing works or can it still be used with risks?**

- Depende de si tiene una fecha de caducidad o no
  - o Fecha de caducidad: software con maps que hay que estar actualizando cada año
  - o Sin fecha de caducidad: Productos IoT para la casa que pueden interferir entre ellos
- Posibles problemas en cuanto a legalidad

**do you have any data on the lifespan of a software project? How long does it typically last?**

- Depende del tipo de empresa
  - o Producto
  - o Servicio

Si tienes un contenedor, es muy importante saber lo que metes dentro, sobre todo, identificar la parte crítica de tu empresa, de tus datos y no delegar en terceros esa información.

Anderson dice que estaría bien que hubiese un estándar sobre el mantenimiento del árbol de dependencias, de manera que las librerías tengan metadatos y cuando se supiese de alguna que estuviese infectada o fuese vulnerable, te avisase automáticamente de que la estás utilizando y en qué paquete, etc..

### Ciclo de vida del software

Inicio proyecto – Desarrollo – Mantenimiento – Mantenimiento final – Obsolescencia

El momento más caro del desarrollo del software es el mantenimiento cuando está muy avanzado

Para ser un buen Senior es vital saber el contexto en el que se desarrolla el software, el modelo de negocio de la empresa, las opciones... => No existe el desarrollador senior que simplemente programa

Pensar el tiempo de vida útil que tendrá el software que tienes que desarrollar hará que tomes mejores decisiones, por ejemplo si tienes que mantener la aplicación durante 10 años, no vas a escoger de lenguaje de programación un lenguaje reciente, escogerás uno que tenga mucha comunidad y que ya lleve muchos años como java.

Debate en la UE sobre cuanto mantener el software, mínimo 2 años

## Bibliografía

- Historia del ataque a Solarwinds: <https://www.wired.com/story/the-untold-story-of-solarwinds-the-boldest-supply-chain-hack-ever/>
- Wikipedia [https://es.wikipedia.org/wiki/Obsolescencia\\_programada#](https://es.wikipedia.org/wiki/Obsolescencia_programada#)
- Blog de un tio random <https://ramonramon.org/blog/2014/08/19/obsolescencia-del-software/>
- Blog random <https://udax.edu.mx/experiencia/tecnologia-y-software/explorando-los-ecosistemas-de-software-en-el-mundo-digital-actual>
- Openwebinars <https://openwebinars.net/blog/arquitectura-de-software-que-es-y-que-tipos-existen/>
- Mantenimiento software <https://fastercapital.com/es/contenido/Costo-del-software--El-coste-del-mantenimiento-del-software.html>
- Aspectos legales obsolescencia ue: <https://eur-lex.europa.eu/legal-content/ES/TXT/?uri=CELEX%3A32019L0770>
- Aspectos legales obsolescencia españa [https://cec.consumo.gob.es/CEC/comunicacion/publicaciones/docs/Folleto\\_Garantia\\_Legal\\_Minima.pdf](https://cec.consumo.gob.es/CEC/comunicacion/publicaciones/docs/Folleto_Garantia_Legal_Minima.pdf)