Group 09: Sergio Llenderrozos Piñera, Pablo Lobeto Arenas and Sergio Rodríguez García

# Software Engineering Radio 559: Ross Anderson on Software Obsolescence

## Introduction

In this episode of Software Engineering Radio, host Priyanka Raghavan interviews Ross John Anderson, Professor of Security Engineering at the University of Cambridge, about software obsolescence. The two discuss the risks associated with software going obsolete and consider several examples of software that has become obsolete over time.

## Understanding Software Obsolescence

Software obsolescence occurs when software is no longer supported by its vendor or when it is no longer compatible with newer hardware or software. This can happen for a number of reasons, including:

- **The vendor may go out of business.**
- **The software may not be profitable enough for the vendor to continue to support.**
- **The software may be incompatible with newer hardware or software.**
- **The software may contain security vulnerabilities that are no longer being patched.**

## Real-World Implications

Software obsolescence can have a number of real-world implications, including:

- **Security risks:** Obsolete software may contain security vulnerabilities that are no longer being patched. This can make it vulnerable to attacks.
- **Compatibility issues:** Obsolete software may not be compatible with newer hardware or software. This can make it difficult or impossible to use the software.
- **Increased costs:** Businesses may have to spend money to upgrade or replace obsolete software.

## Impact of Obsolescence

The impact of software obsolescence can vary depending on the specific software and the organization that is using it. However, in general, software obsolescence can lead to:

- **Increased security risks**
- **Compatibility issues**
- **Increased costs**
- **Reduced productivity**
- **Lost revenue**

## Challenges and Solutions

There are a number of challenges associated with managing software obsolescence. These challenges include:

- **Identifying obsolete software:** It can be difficult to identify all of the obsolete software that an organization is using.
- **Prioritizing upgrades:** It can be difficult to prioritize which obsolete software should be upgraded first.
- **Funding upgrades:** Upgrading or replacing obsolete software can be expensive.
- **Finding compatible replacements:** It can be difficult to find compatible replacements for obsolete software.

There are a number of solutions to the challenges of managing software obsolescence. These solutions include:

- **Developing a software obsolescence management plan:** A software obsolescence management plan should identify the organization's goals, risks, and resources. It should also outline a process for identifying, prioritizing, and upgrading obsolete software.
- **Using software inventory tools:** Software inventory tools can help to identify obsolete software.
- **Prioritizing upgrades based on risk:** Upgrades should be prioritized based on the risk that the obsolete software poses to the organization.
- **Budgeting for upgrades:** The organization should budget for software upgrades.
- **Evaluating replacements carefully:** Replacements for obsolete software should be evaluated carefully to ensure that they are compatible with the organization's needs.

## Using AI

Artificial intelligence (AI) can be used to help manage software obsolescence. AI can be used to:

- **Identify obsolete software:** AI can be used to scan an organization's network for obsolete software.
- **Prioritize upgrades:** AI can be used to prioritize upgrades based on the risk that the obsolete software poses to the organization.
- **Recommend replacements:** AI can be used to recommend replacements for obsolete software.

## Regulations

There are a number of regulations that can affect software obsolescence. These regulations include:

- **Data protection regulations:** Data protection regulations, such as the General Data Protection Regulation (GDPR), require organizations to

protect their data. This can include using up-to-date software that is free from security vulnerabilities.
- **Industry-specific regulations:** Some industries have specific regulations that require organizations to use up-to-date software. For example, the healthcare industry has regulations that require organizations to use software that is certified by the Health Insurance Portability and Accountability Act (HIPAA).

### Conclusion

Software obsolescence is a serious problem that can have a number of negative consequences for organizations. However, there are a number of steps that organizations can take to manage software obsolescence and reduce its risks.

### Additional Notes

- The episode also discusses the concept of software "deprecation," which is when a vendor announces that it will no longer support a particular piece of software.
- The episode also discusses the importance of knowing what software is compiled with, as this can affect the ability to upgrade the software.
- The episode also discusses the costs of software obsolescence, which can include the cost of upgrading or replacing software, the cost of lost productivity, and the cost of lost revenue.
- The episode also discusses the importance of considering external code when managing software obsolescence, as this can affect the compatibility of the software.
- The episode also discusses the importance of using AI to help manage software obsolescence.
- The episode also discusses