



Universidad de Oviedo

SOFTWARE OBsolescence

GROUP 9

SERGIO RODRIGUEZ GARCIA

PABLO LOBETO ARENAS

SERGIO LLENDERROZOS PIÑERA - UO283367

CONTENT

- 
- 01** MEET ROSS ANDERSON
 - 02** UNDERSTANDING SOFTWARE OBSOLESCENCE
 - 03** REAL-WORLD IMPLICATIONS
 - 04** IMPACT OF OBSOLESCENCE
 - 05** CHALLENGES AND SOLUTIONS
 - 06** POOR MAINTENANCE EXAMPLE
 - 07** GENERAL COMPLICATIONS
 - 08** SOME ARE EASIER TO UPDATE
 - 09** SOFTWARE "DEPRECATION"
 - 10** KNOWING WHAT YOU COMPILE
 - 11** COSTS
 - 12** EXTERNAL CODE
 - 13** DECISION MAKING
 - 14** USING AI
 - 15** REGULATIONS

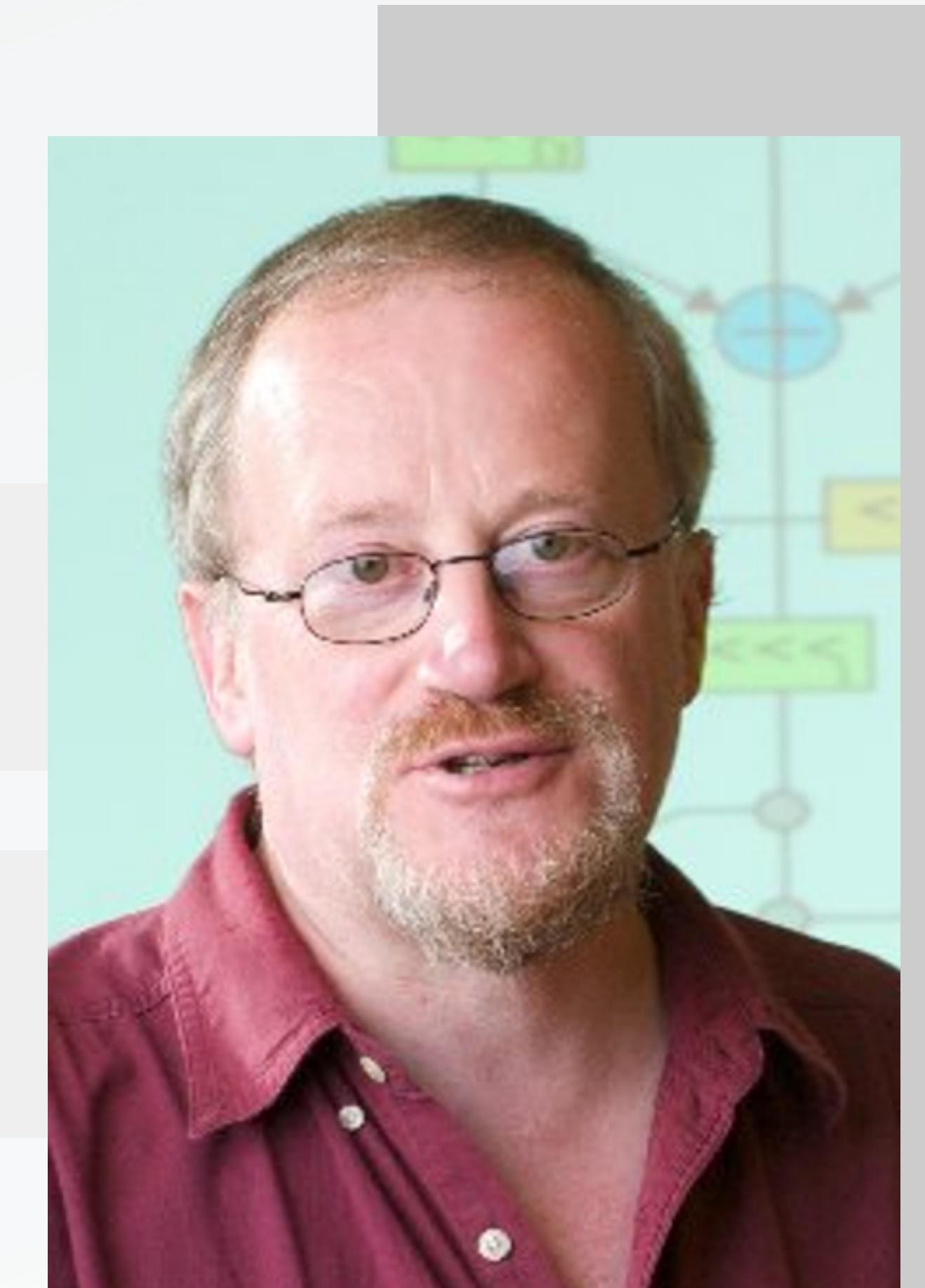
MEET ROSS ANDERSON



University of Cambridge, University of Edinburgh



Author of "Security Engineering, A Guide to Building Dependable Systems"



UNDERSTANDING SOFTWARE OBsolescence

DEFINITION

Software evolution leading to dependability issues and vulnerabilities

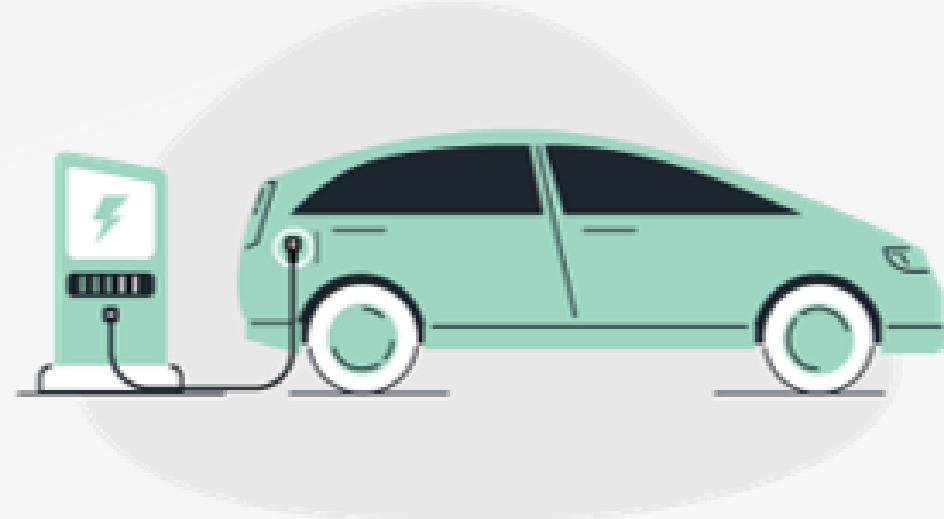
COMPLEXITY

Impact of software ecosystems and machine learning integration

EXAMPLE

Smart devices becoming dysfunctional due to discontinued support

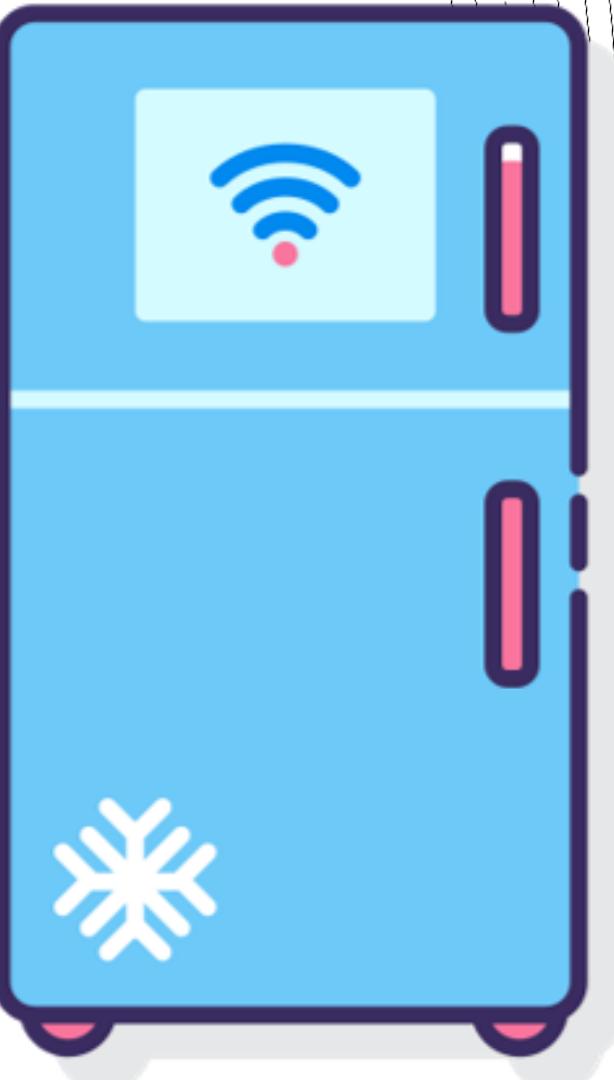
REAL-WORLD IMPLICATIONS



- Transition to electric cars and employment dynamics
- Challenges in coordinating vulnerability disclosure and patching
- Need for better tools and institutional redesign

IMPACT OF OBSOLESCENCE

- Can obsolete software still be used?
- Maintenance challenges beyond software: Devices embedded with software

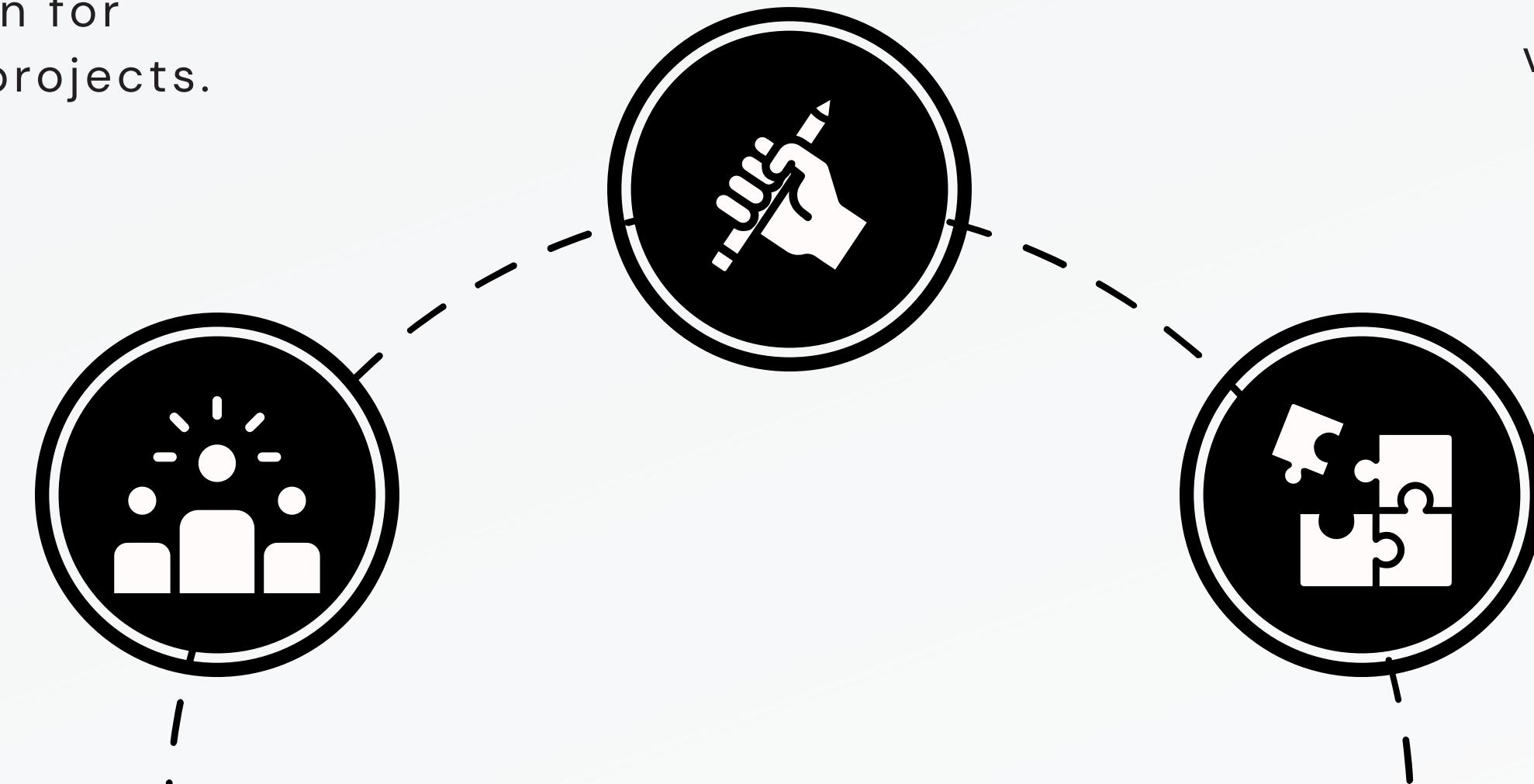


CHALLENGES AND SOLUTIONS

Lack of definitive lifespan for software projects.

Software maintenance practices in different sectors

Legal and regulatory efforts to address systemic vulnerabilities in IoT devices.



POOR MAINTENANCE EXAMPLE



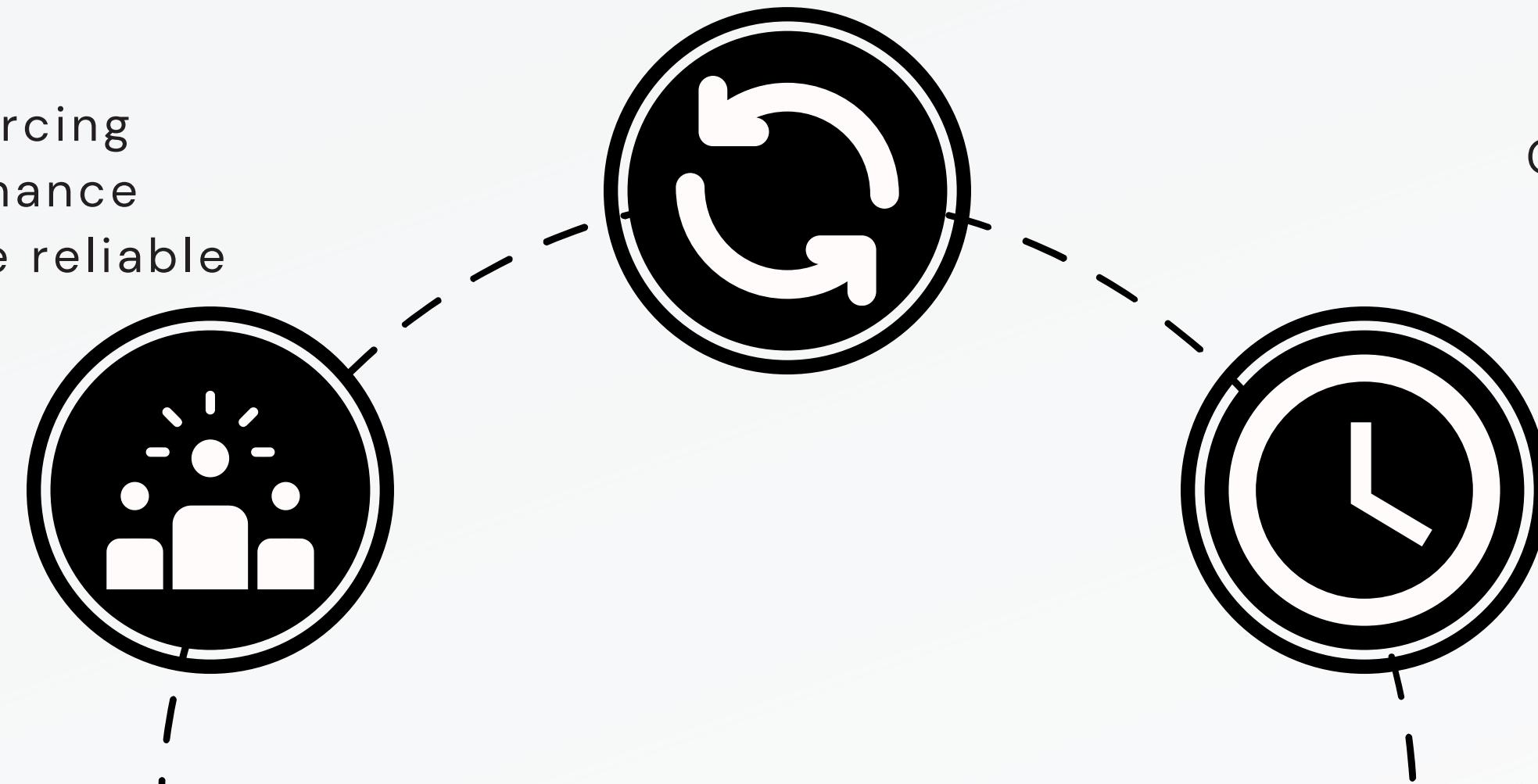
- CCTV cameras from xiaomi were used for DDOS attacks, and the software couldn't be upgraded
- Solar wind used to be a great sofware company but it was bought and the great engenears were replaced by low-cost labor.

GENERAL COMPLICATIONS

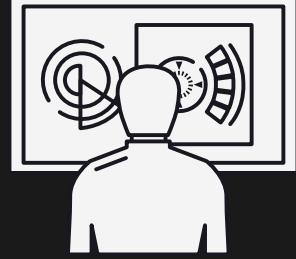
Old software may be very expensive
to replace

Outsourcing
maintenance
may not be reliable

Customers desire
products with
longevity.



SOME ARE EASIER TO UPDATE



Control system are being reparametrice into just one gateway between them and the internet



A car involves numerous software components, most of which cannot be updated. However, if they were capable of receiving updates via the internet, it would be imperative to ensure security

SOFTWARE “DEPRECIATION”

- Software needs to be maintain with small tweaks, by refactoring or by exchanging components
- Maybe for a system a big update is changing a component but for the developer of that component is a small tweak

KNOWING WHAT YOU COMPILE

It is important to know what code is compiling in critical systems that may skip the security parameters

In an ideal future, when a vulnerability in a library is discovered, there would be a button that indicates where it is being affected

This will initially be accomplished by manual checking, but large companies could develop automatic tools.

COSTS

Software is continuously evolving according to the market and client needs.

We must understand what is behind developing software.

90%



EXTERNAL CODE



If breaks, possibility of putting people into the community and fix the problem.

OPEN SOURCE



Be carfeul with negotiations about support and money

COMPANIES

DECISION MAKING

What programming language and tools are you going to use?

- Costs
- Risks
- Machine learning



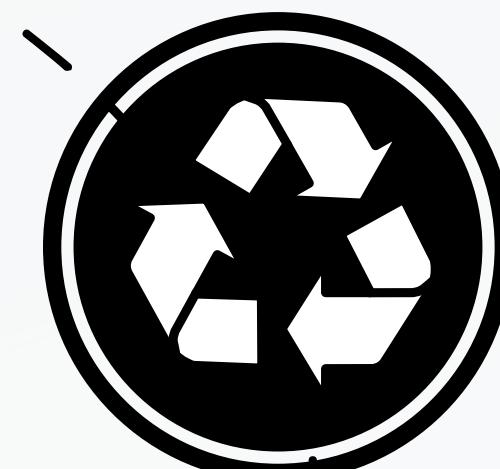
• Where are you going to invest your own time and expertise?

Career bets



Are you going to become a first-class front-end programmer?

- Maintainability
- Innovations



USING AI

Increasingly, AI is entering our lives and, unsurprisingly, software. This brings both advantages and disadvantages

Using AI for software development provides productivity, faster time-to-market, and improved code quality.

PRODUCTIVITY

They can also mitigate challenges associated with developing and maintaining software

CHALLENGES

AI alone cannot replace the expertise and judgment of human developers. Quality? Security?

RISKS



REGULATIONS

Sustainability is a real big deal.

Who's going to regulate the additional costs of maintaining software?

Every product with software has to be patch its software for at least two years

??%



**THANK'S FOR
WATCHING**



Universidad de Oviedo