Exercice 1 : ***Study of the Failure Case of "Ariane 5" Due to a Coding Error***

The crash of Ariane 5 resulted from a series of technical and strategic errors. The computer in question was responsible for providing data during the flight control. It had been developed for Ariane 4 and had successfully completed several flights.

The strategic errors stemmed from the fact that the computer was reused as is for Ariane 5, without any revision of the specifications or additional testing. Unfortunately, the input data for Ariane 5 was completely different from that of Ariane 4. This would lead to the bug that resulted in the destruction of the launcher.

A second computer operated concurrently and in parallel with the first; however, since its design was identical to the first, the same causes produced the same effects.

Technical Errors:

1. The code simply contained an assignment from a 64-bit data variable to a 16-bit data variable. Since the input data for Ariane 5 was larger than expected, it sometimes happened that a value greater than 16 bits was assigned to the variable coded on 16 bits. An exception was then logically raised.
2. In Ada language, a mechanism exists to handle this type of exception. Here, no protective mechanism was provided. The exception was passed during flight control, which treated it like any other value provided by the computer. This led to its aberrant behavior, resulting in the loss of trajectory and then triggering self-destruction.

Questions:

1. What lessons can we draw from this case?
2. The development of industrial or advanced systems is often complex and requires significant time for success. What are the most suitable models for such projects, and why?

Solutions:

1. The lessons we can draw from this are:
   - We need to adapt the software to the new hardware and system.
   - Review and update the requirements document to align with the new hardware specifications.
   - Never skip integration testing, even if it worked perfectly on another system.
   - It's important to manage exceptions and handle them appropriately.
   - A risk of zero does not exist there is always a chance, no matter how small, that a serious bug might occur only during the execution of the software (Vicious Bug).

2. The most suitable models for such projects are:
   - **Spiral**: Suited for complex projects and focuses on risk management.
   - **Incremental**: Breaks complex projects into modules, making it easier to debug and test each module independently and integrate them minimizing errors and bugs.
   - **Prototyping**: Suited for innovative projects that contain both hardware and software components.
   - **Hybrid**: Breaks the project into different parts, each with its suitable life cycle, offering the best flexibility and benefits from each life cycle.

Exercise 2: "Choosing an IT Solution"

A research laboratory wants to equip itself with an integrated management system for its research teams to handle information related to researchers, their scientific productions in conference papers, the number of journal articles published, the number of research trips, and scientific leave.
To achieve this, three possible approaches are available to the laboratory:

1. Acquire a database management system and develop its own system based on this tool.
2. Purchase a similar system from another laboratory and adapt it to its own needs.
3. Join a group of laboratories, establish a common requirements document, and contact a software vendor to develop a single system for all.

Question: Identify two possible risks for each of these strategies and propose risk resolution techniques that would help decide which approach to adopt.

Solution :

Risks of each solution :

1st Solution

- Risk of insufficient budget to acquire a database management system.
- Potential development challenges due to lack of expertise.
- Risk of extended development time.

2nd Solution

- Risk of insufficient budget to purchase the software.
- Challenges in adapting the software due to limited expertise or lack of customizability. =>maybe acquiring new software that meets the lab's requirements is better than to adapt an existing one.