**1-Adoption of the secure coding standards and not leaving security at the end**

Do not leave security at the end means considering security at the left in the development process. Security must be integrated into each part of the software development life cycle to avoid colossal repair costs, delays, and bad software quality. In other words, the traditional software development life cycle must begin with security attached to each phase to reduce vulnerability surface. In the context of agile and cloud computing, shifting security left improves development productivity and raises serious security concerns.

**2-Evaluation and assessment of risks and cost-benefit of mitigation**

In continuous integration, continuous deployment, automation is the key. Automation tools introduced early in the development stage detect the vulnerabilities and allow developers to mitigate eventual threats. Developers need to know what static analysis tool they need based on their threat modeling.

In addition to the source code, automation tools also help check dependencies and third-party components, reducing colossal repair costs and improving software quality.

**3- Zero trust**

The concept of zero trust improved the traditional security model by isolating and addressing each threat vector. In terms of infrastructure, the reverse proxy configurated between the internet and the server and the single sign-on gateways supports the model. The traditional triple-A is not enough to grant access because the zero trust considers everything external such as people, devices, applications, networks, data must be checked continuously. In addition, with visibility and analytics, we can see and monitor users' privileges using the principle of least privilege. The concept is adapted to our digital workspace, but the well-implemented zero trust requires experts. The concept shifts the security perimeter from the corporate business to the user, enhancing the user experience.

**4-Implementations and recommendations of security policies**

The security policies need to be planned for its implementation. Developers must be trained on how to use the different tools. The policy must adhere to the defense in depth and least privilege principles. It is good practice to update the new threats and covers other languages.