* **Adoption of a secure coding standard, and not leaving security to the end.**  
  Adopting a secure coding strategy result in a more secure application. By not leaving security until the end a developer can reduce development cost, development time, and more secure applications.
* **Evaluation and assessment of risk and cost benefit of mitigation.**  
  Using a static analysis tool to find potential risks, coding standard rule violations and using the risk assessment from rules that are violated to prioritize which ones have the highest priority. The cost to repair damages caused by a successful breach or intrusion could easily total into the millions of dollars, so, evaluating and using risk assessment to ensure code meets all standards can have huge cost benefits.
* **Zero trust**  
  Zero trust can go a long way in protecting an application from being attacked. While it is not an easy system to setup but if software is developed utilizing such things as SSO it only helps to increase the security of software. Zero trust is a paramount in securing company resources that need to be accessed by remote users.
* **Implementation and recommendations of security policies**  
  Creating standardized company policies in relation to software development is almost a requirement in today’s world. Hackers are only growing more sophisticated which makes securing developed software even more important. Implementing and recommending security policies to which developers can follow is just another layer in a multi-layered approach to security.