Overall, I think it is important to adopt a secure coding standard for a number of reasons not just related to security. I think a good secure coding standard can help new (and older) programmers find a consistent way of building and maintaining code. While the security of the standard is definitely important, using the standard helps make code more maintainable and reliable as ever programmer is on a relatively similar page. It also helps to improve code review as there are key points that are easily recognizable in code rather than a line by line reading of the code and considering every possible implication of every line in depth. A secure standard can be a starting point for solving more common problems as well as those problems are likely outlined in the standard (with examples) if they can affect security, making an internal resource a first stop for developers.

Evaluation and assessment of risk and cost benefit of mitigation is important because it takes the theoretical into practical application. A security standard may make sense in theory, but if it is difficult or convoluted to implement it likely to be ignored or expose new vulnerabilities. Evaluation and assessment of risk helps an organization (or individual) understand where potential threats lie and how feasible they are, which allows a coding standard to be mutated to meet the needs of the risk if warranted. Additionally, the cost benefit of mitigation can guide large scale decisions. An example would be migrating to a new hosting service (maybe a cloud provider) and doing a cost benefit analysis. Maybe this move will improve security overall but would exceed the value provided within a couple years based off the cost, or the cloud provider does not allow a concrete to monitor activity (which may offset the security improvements entirely as if there is an intrusion you may have no way to tell how it happened or how to mitigate the damage). Determining the cost and benefit of decisions in the context of security is useful because it can lead to long term security advantages or disadvantages, and some of these disadvantages may be mitigatable with a proper analysis.

Zero trust is becoming an ever more important aspect of security in the modern working environment. Not trusting anything, whether it is on your network or not enables you to have greater control over your network and an improved ability to detect malicious actors. As working becomes a more remote endeavor, network security is becoming significantly more difficult without the zero-trust model. For every employee working remote that is one more opportunity and vector of attack if their connection with your backend systems (or public systems) is not properly handled. The zero-trust model allows a single authenticatable connection between each system and the user, this means that a connection can be more effectively monitored for malicious actions and quickly removed if determined to be a threat or bad actor.

Implementing a security policy is absolutely necessary for almost any conceivable organization in this increasingly connected world. It is also important the policy is flexible and mutable as the world changes. The most secure protocols in the world could instantly become a liability with the discover of a vulnerability, and as such the policy needs to be able to quickly mitigate these vulnerabilities and provide the same level of security at a minimum. Additionally, it is important to consider outside perspectives (whether internally or externally generated) as the one thing consistent in software is that nobody knows everything. Everyone has blind spots and if security is primarily left on the shoulders of the Security or IT team there is a good chance that an improvement will be missed, or an opportunity lost. While these teams are focused on security it is one everyone to help improve and evaluate those security policies.