Four Approaches to Security Champions Programs

Security Champions programs can drive secure development, but success depends on design and execution. Below are four real-world scenarios highlighting what works and what doesn't.

Company 1. Checkbox Security (FAIL)

This company introduced a Security Champions program primarily to demonstrate progress to stakeholders. On paper, it existed. There were a lot of guidelines and security checklists on many stages of development process. In practice:

- Employees were labeled as "champions" without clarity
- No training, guidance, or time allocation was provided
- There was no budget or support structure

The program had no clear expectations or outcomes. It served more as a compliance or marketing tool than a real initiative.

Company 2. No Time, No Metrics (FAIL)

This company made an initial push:

- Some training sessions were organized
- A basic reward system (e.g., recognition or certificates) was introduced

However, the effort lacked operational integration:

- Developers were overwhelmed with existing workloads
- There was no system for measuring engagement or outcomes
- The only offer was occasional training; no hands-on support, coaching, or check-ins

Without accountability or KPIs, enthusiasm faded. Over time, fewer people participated, and the program lost visibility until it became inactive.

Company 3. Retention problem (FAIL)

Here, the program launched with strong initial energy:

- Champions received access to learning resources and events
- Awards and recognition were offered, boosting motivation early on

The initiative depended on a few engaged individuals. When they left the company, the program stalled:

- No formal documentation or process to replace outgoing champions
- Knowledge and momentum weren't transferred

The company attempted to restart the program recently, but still retention was a problem.

Company 4. Structured and Measurable (SUCCESS)

This company built a mature and effective Champions model:

- Security was seen as a shared responsibility, not siloed
- Champions were selected based on interest and engagement, not title or role
- A structured training path supported both beginners and advanced participants
- Champions contributed to building security tools and organizing internal security events
- Secure Software Development Lifecycle (SSDLC) processes were actively used
- Metrics tracked multiple dimensions: number of threat models created, vulnerabilities found internally and externally, training completed, and developer effectiveness
- A level based champion system was in place, allowing for progression and differentiated responsibilities

The result was a program that actively contributed to security improvements and team awareness, and which became a long-term part of the organization's engineering culture. Even employee retention didn't ruin it because of standarized way of working.

Conclusion

Security Champions programs succeed when they are thoughtfully designed, with dedicated support, measurable goals, and real integration into daily work. They fail when reduced to branding, symbolic roles, or one-off efforts. Sustainability comes from investment, structure, and relevance to those involved.