Cybersecurity Dynamics in Software Development Environment: What system traps do exist?

Sander Zeijlemaker and Michael von Kutzschenbach

1. Increasing hacking efforts

- Adaptation trap: decay of coding and testing capability impacts future discovery and solving efforts.
- Capability trap: low efforts on discovery and solving evoke future problems.
- Decision trap: no overview on overall software base quality due to short term cycles evoke crappy software generation
- Acceptance trap: mistakenly accept known vulnerabilities (no solving) evoke future successful hacks
- Attacker defender interaction: arms race between attack and defender

2. Raise concerns about policies countering this effect

- **Delivery method**: Traditional vs Agile Project Method
- **Software delivery**: Minimal errors vs Fast delivery
- **Delivery priority**: Maximum reliability vs Minimal Viable Product
- **Managerial decision-making dilemma**: available software yield income while working on software generate costs

3. As the SDLC eco-system provide us five different system traps

- **Trap details**:
  1. Adaptation trap: decay of coding and testing capability impacts future discovery and solving efforts.
  2. Capability trap: low efforts on discovery and solving evoke future problems.
  3. Decision trap: no overview on overall software base quality due to short term cycles evoke crappy software generation
  4. Acceptance trap: mistakenly accept known vulnerabilities (no solving) evoke future successful hacks
  5. Attacker defender interaction: arms race between attack and defender

Authors

Sander Zeijlemaker RA RE MSc SCF, PhD student Radboud University, IMR Faculty Nijmegen
Prof. Dr. Michael von Kutzschenbach, University of Applied Sciences and Arts Northwestern Switzerland
FHNW, Institute of Management