

 OWASP GLOBAL **AppSec**

ViE'26
NNA JUN
25-26

25

years
of open source security

OWASP GLOBAL AppSec

ViE'26
NNA JUN
25-26

25

years
of open source security

Keep It Between Us: Manipulating Humans for Better AppSec (Ethically)

BY: Nariman Aga-Tagiyev

Governance	Design	Implementation	Verification	Operations
Strategy and Metrics Create and promote Measure and improve Stream AStream B	Threat Assessment Application risk profile Threat modeling Stream AStream B	Secure Build Build process Software dependencies Stream AStream B	Architecture Assessment Architecture validation Architecture mitigation Stream AStream B	Incident Management Incident detection Incident response Stream AStream B
Policy and Compliance Policy & standards Compliance management Stream AStream B	Security Requirements Software requirements Supplier security Stream AStream B	Secure Deployment Deployment process Secret management Stream AStream B	Requirements-driven Testing Control verification Misuse/abuse testing Stream AStream B	Environment Management Configuration hardening Patch and update Stream AStream B
Education and Guidance Training and awareness Organization and culture Stream AStream B	Secure Architecture Architecture design Technology management Stream AStream B	Defect Management Defect tracking Metrics and feedback Stream AStream B	Security Testing Scalable baseline Deep understanding Stream AStream B	Operational Management Data protection Legacy management Stream AStream B

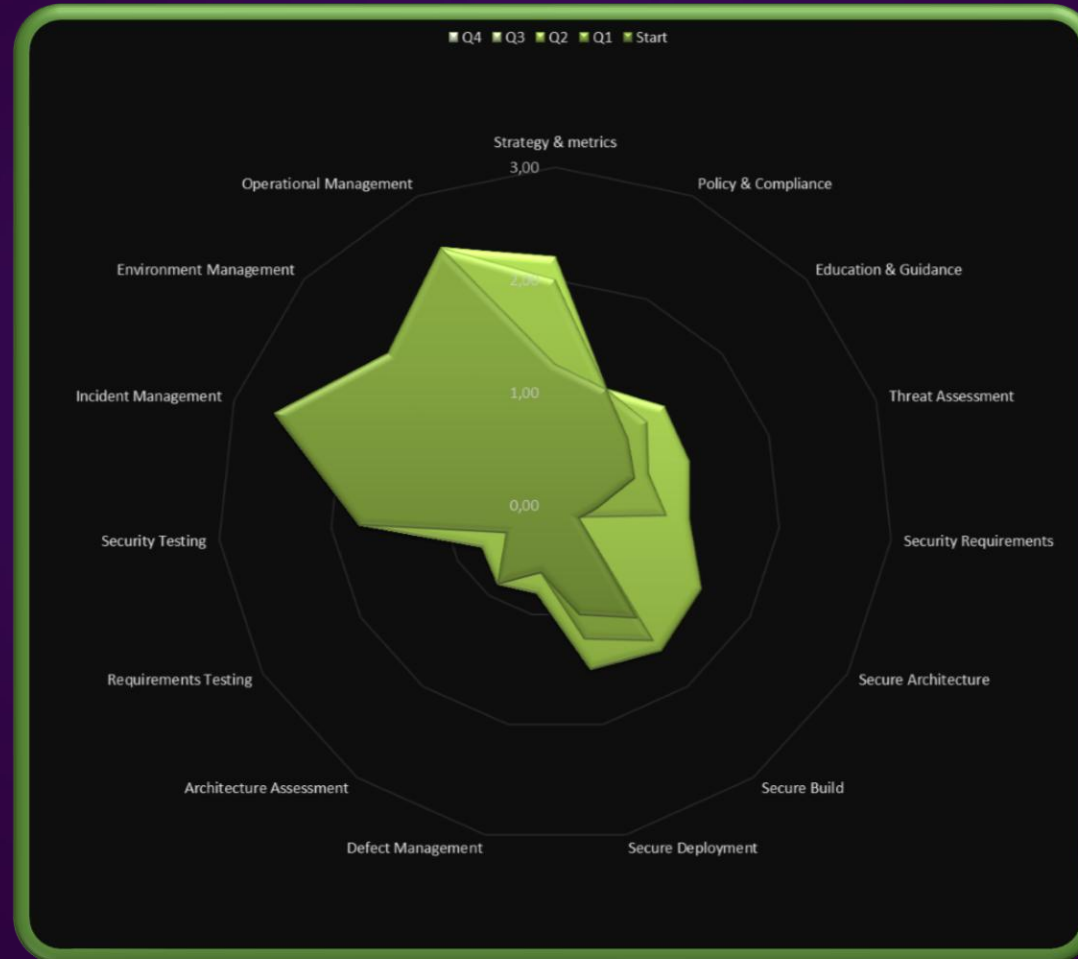


Assess



Set target

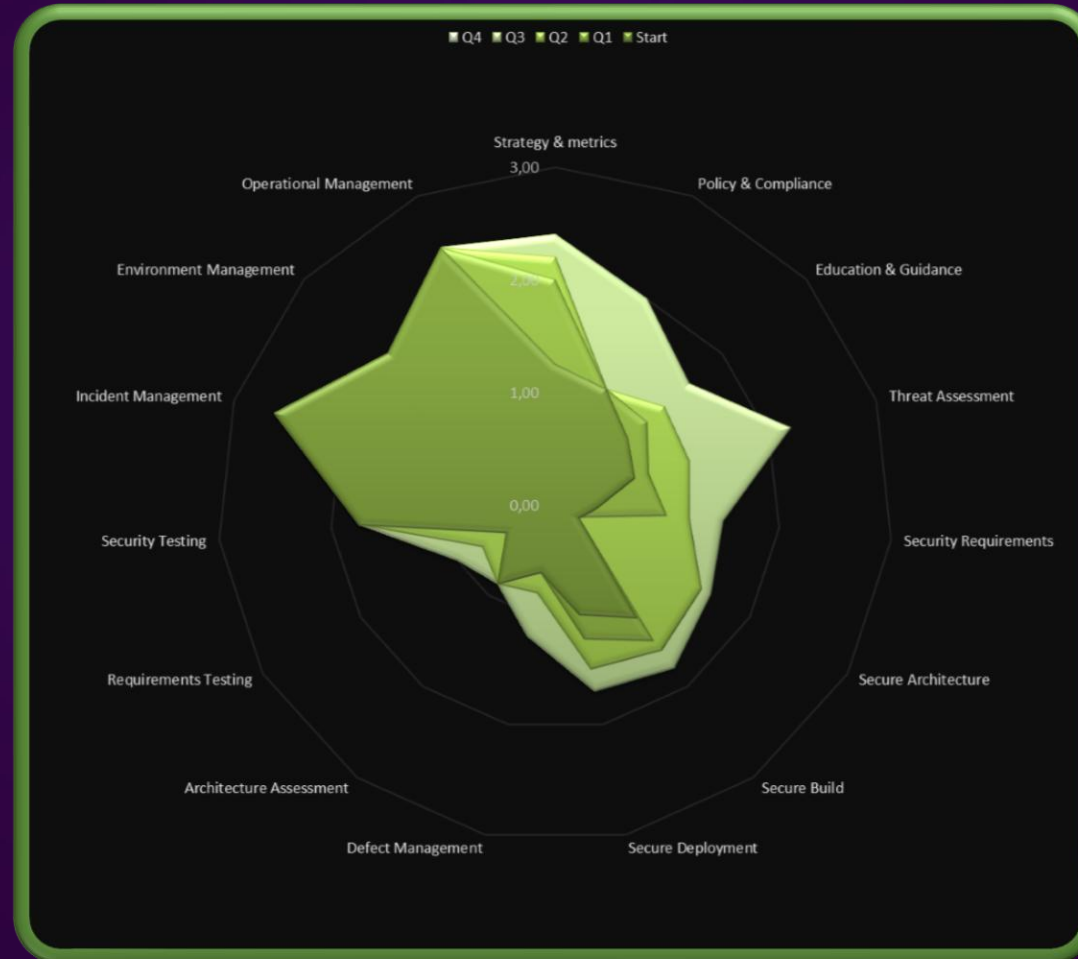




Assess



Set target



Work

Current
state



Can a good engineer get you
from A to B?

Target state



maturity



Requirements,
roadmap defined

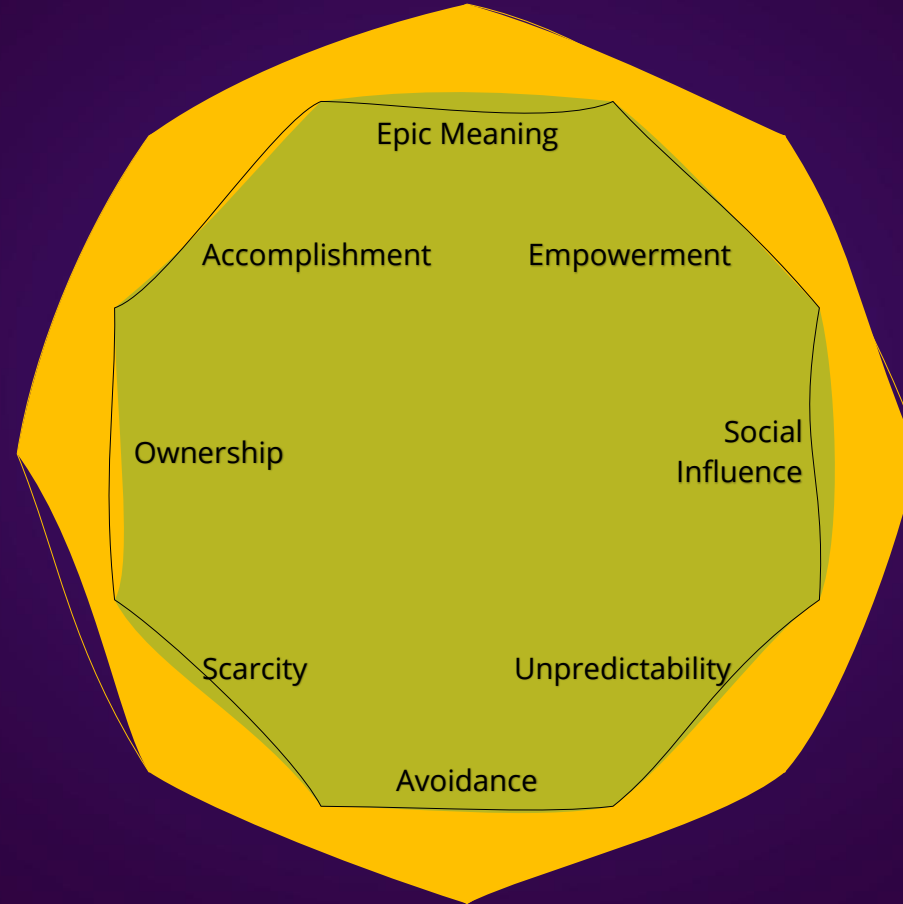




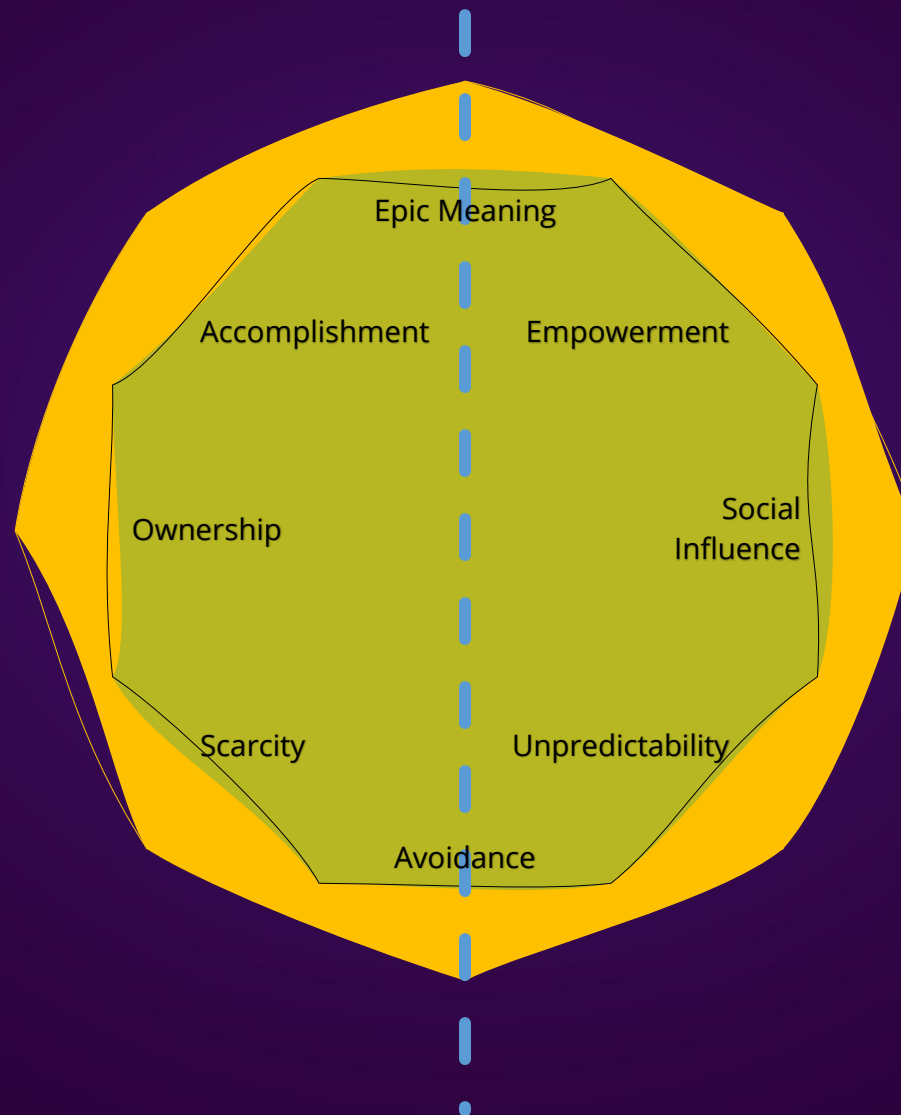
If you know some
tricks...

$$B = M + A + P$$

Behavior Motivation Ability Prompt

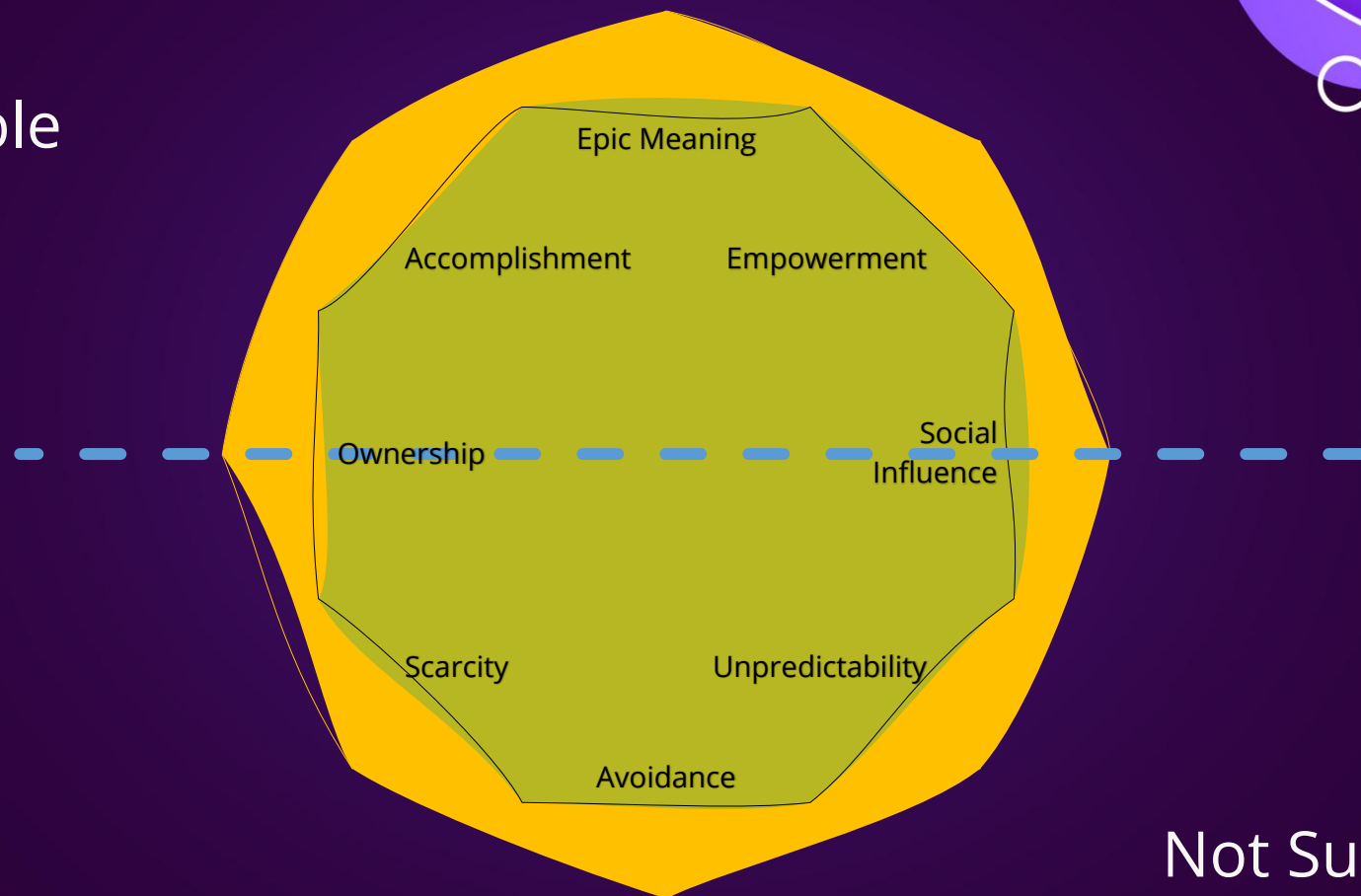


Extrinsic
Depends on
Reward



Intrinsic
Rewarding
In itself

Sustainable
Not Urgent



Urgent
Not Sustainable

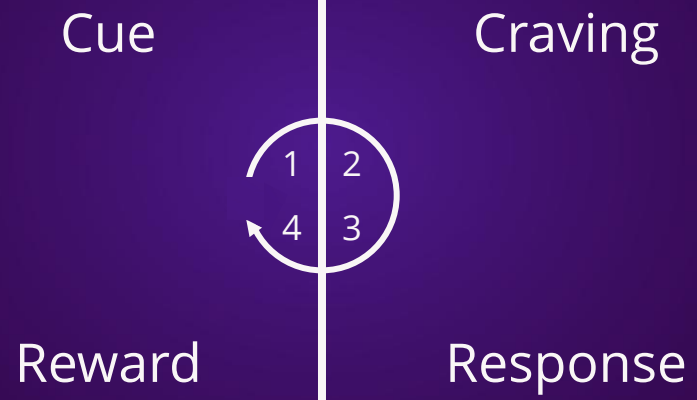
$$B = M + A + P$$

Behaviour Motivation Ability Prompt



With some reinforcement, behaviour
will become a Habit

Charles Duhigg's Habit Loop



Cue (Trigger)

The dependency scanner posts an alert about a vulnerable library version in the project's repository.

Craving (Motivation)

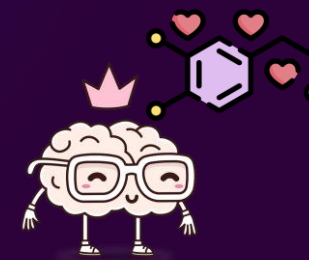
The developer wants a clean, "green" build status and to keep the team's security dashboard free of open vulnerabilities.

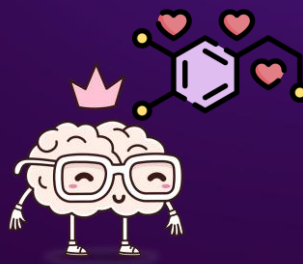
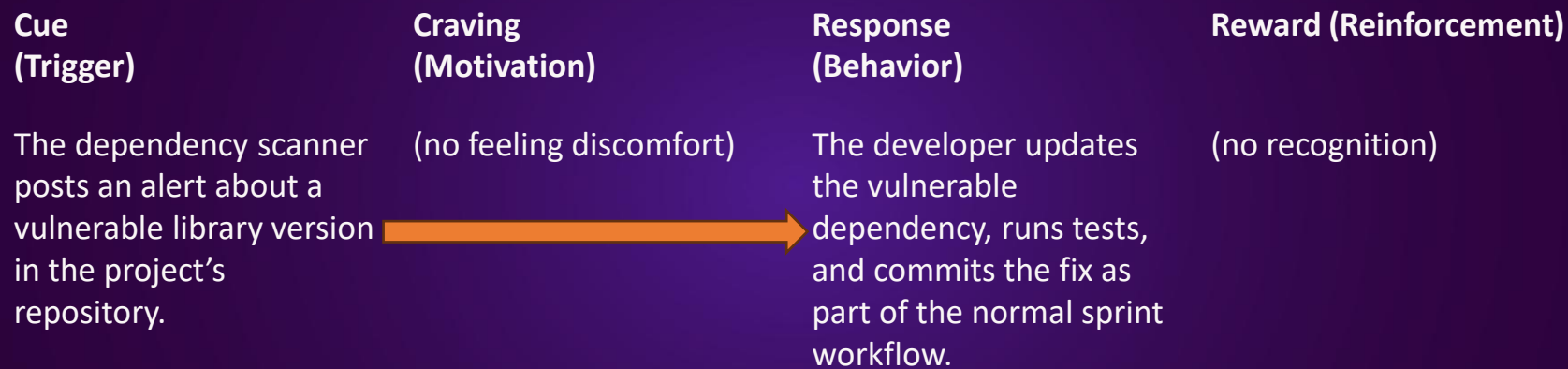
Response (Behavior)

The developer updates the vulnerable dependency, runs tests, and commits the fix as part of the normal sprint workflow.

Reward (Reinforcement)

The alert disappears, the CI/CD pipeline turns green, and the security dashboard shows progress - giving a small sense of accomplishment and recognition from the team.





How to create a
Good Habit?



The 1st Law (Cue)	Make it obvious
The 2st Law (Craving)	Make it attractive
The 3st Law (Response)	Make it easy
The 4st Law (Reward)	Make it satisfying



How to create a
Bad Habit?

Inversion of the 1st Law (Cue)	Make it invisible
Inversion of the 2st Law (Craving)	Make it unattractive
Inversion of the 3st Law (Response)	Make it difficult
Inversion of the 4st Law (Reward)	Make it unsatisfying



Implementation

The invisible



The secret, evil plan
to convert activities
into habits

The visible



Whitepapers, Policies,
Documentation,
Announcements



VIENNA'26 JUN 25-26



THANK YOU!