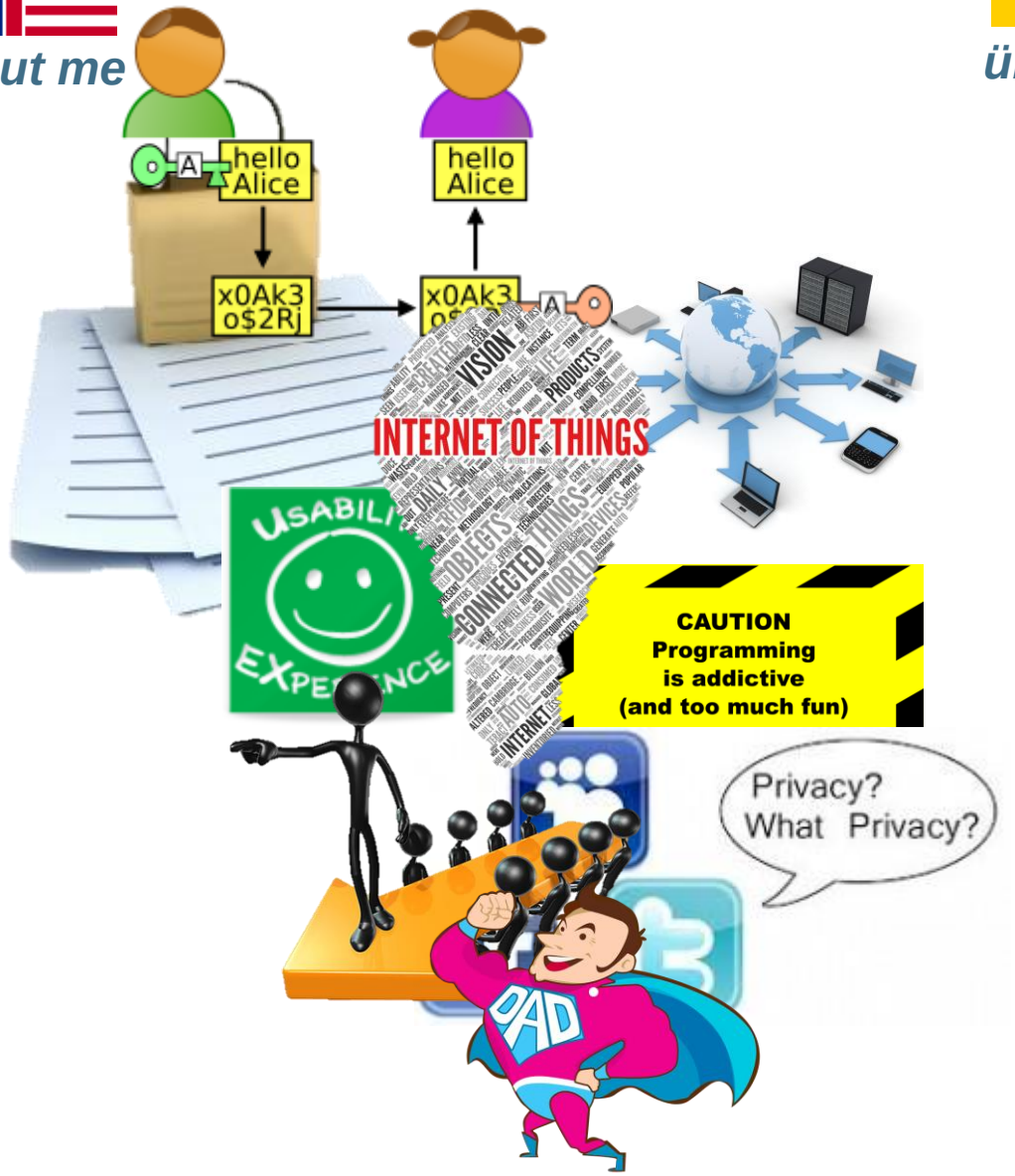
Sicherheit messen: Software und Prozesse

MARK VINKOVITS
LOGMEIN

magamról

about me

über mich



DR. MARK VINKOVITS
MGR APPLICATION SECURITY (EMEA)

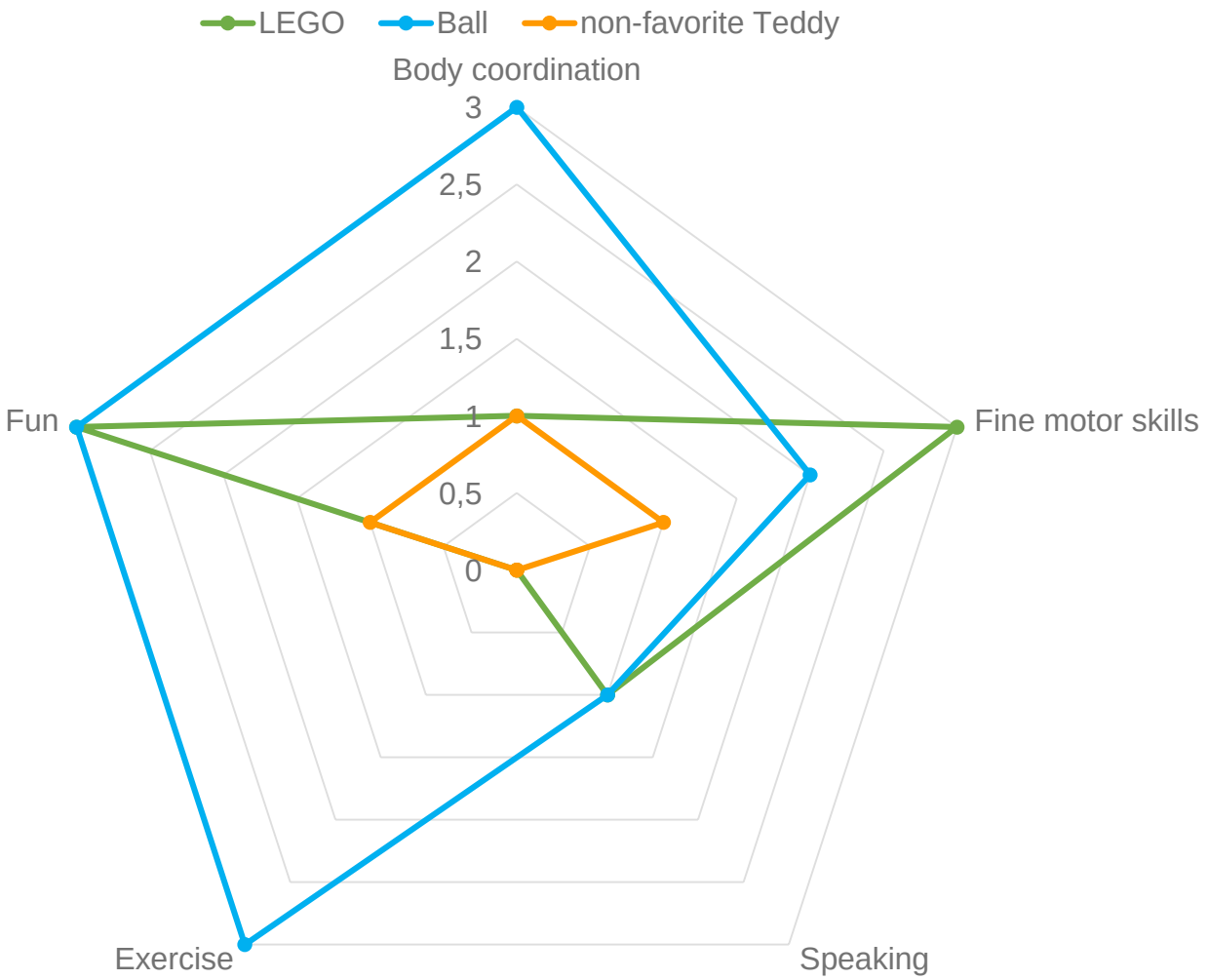
LogMeIn

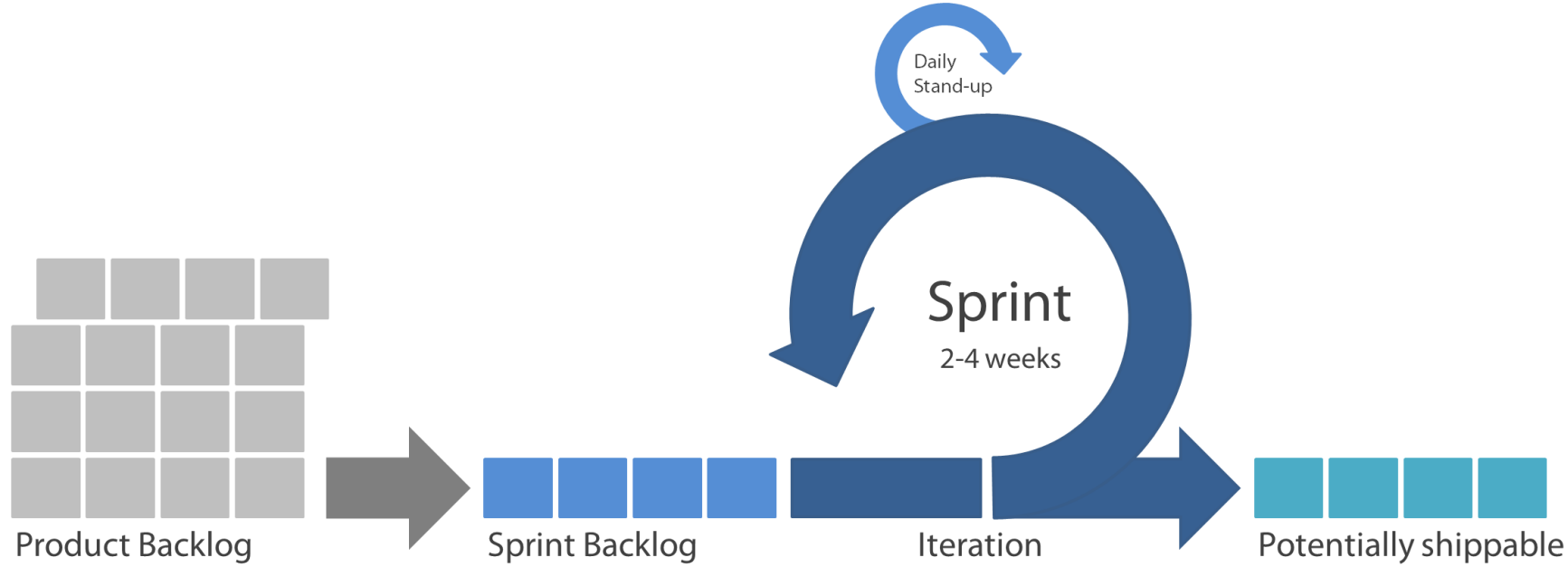
COPRODUCTION WITH ATTILA TÖRÖK, JÓZSEF OTTUCSÁK, SÁNDOR PÁLFY

új nézet *new perspective* *neue Sichtweise*

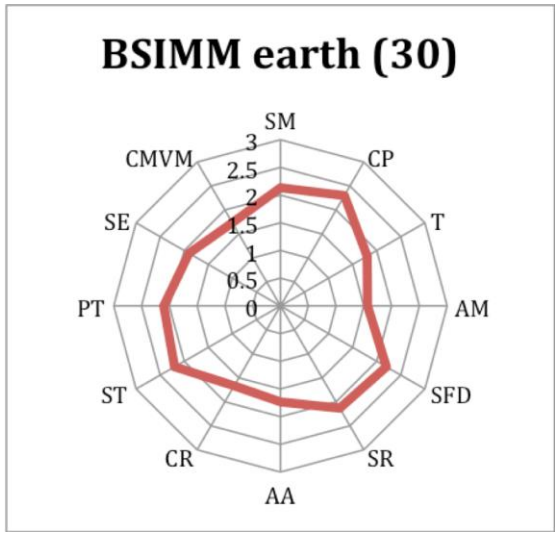
termék menedzser *Product Owner* *Produkt ohne*

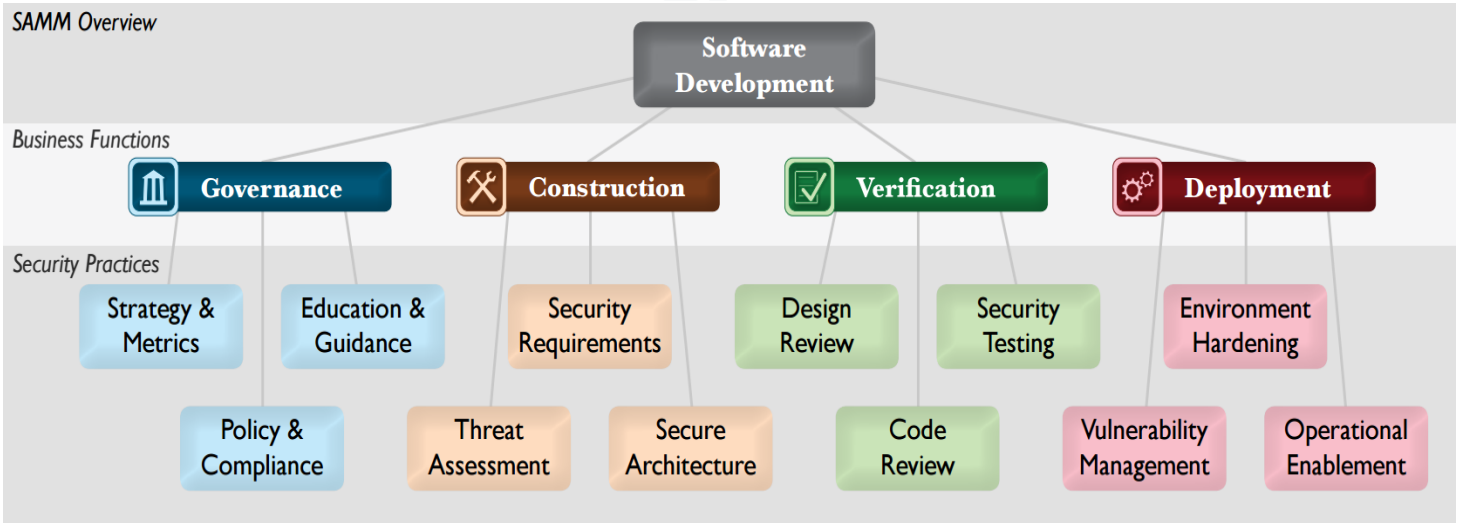

LEGO
LEGO
LEGO
STAR WARS
5.99

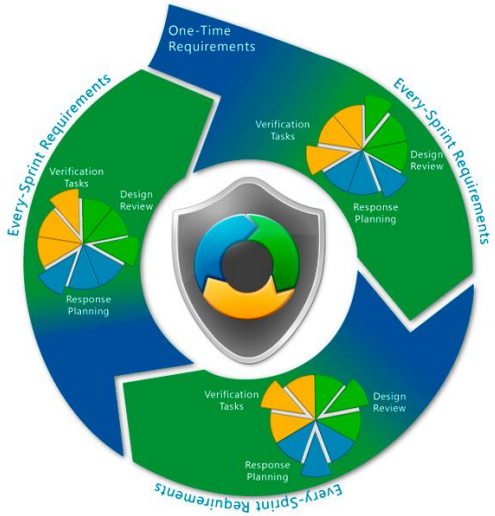

LEGO
Ball
non-favorite Teddy
Body coordination
3
2,5
2
1,5
1
0,5
0
Fine motor skills
Speaking
Exercise
Fun

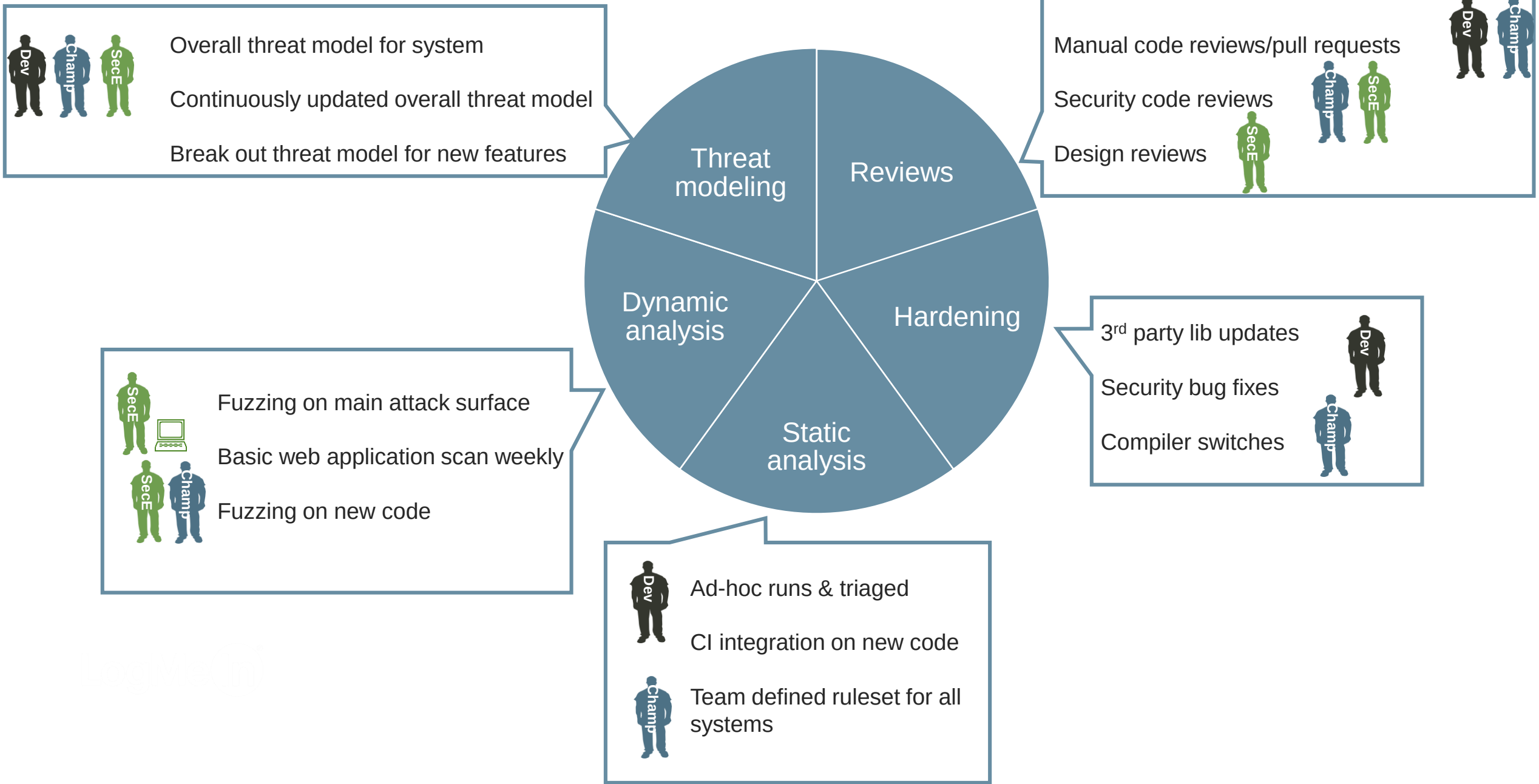

Daily
Stand-up
Sprint
2-4 weeks
Product Backlog
Sprint Backlog
Iteration
Potentially shippable
product increment

VS


BSIMM earth (30)
SM
CP
T
AM
SFD
SR
AA
CR
ST
PT
SE
CMVM
3
2.5
2
1.5
1
0.5
0


SAMM Overview
Software Development
Business Functions
Governance
Construction
Verification
Deployment
Security Practices
Strategy & Metrics
Education & Guidance
Security Requirements
Design Review
Security Testing
Environment Hardening
Policy & Compliance
Threat Assessment
Secure Architecture
Code Review
Vulnerability Management
Operational Enablement


One-Time Requirements
Every-Sprint Requirements
Every-Sprint Requirements
Every-Sprint Requirements
Verification Tasks
Design Review
Response Planning


Dev
Champ
SecE
Overall threat model for system
Continuously updated overall threat model
Break out threat model for new features
Threat modeling
Reviews
Manual code reviews/pull requests
Security code reviews
Design reviews
Hardening
3rd party lib updates
Security bug fixes
Compiler switches
Static analysis
Ad-hoc runs & triaged
CI integration on new code
Team defined ruleset for all systems
Dynamic analysis
Fuzzing on main attack surface
Basic web application scan weekly
Fuzzing on new code

stati[kus|c|sche] anal[íz|ys][is|e]

BSIMM

- **CR1: Use automated tools along with manual review**
- **CR2: Use automated tools with tailored rules**
- **CR3: Enforce coding standards**

OpenSAMM

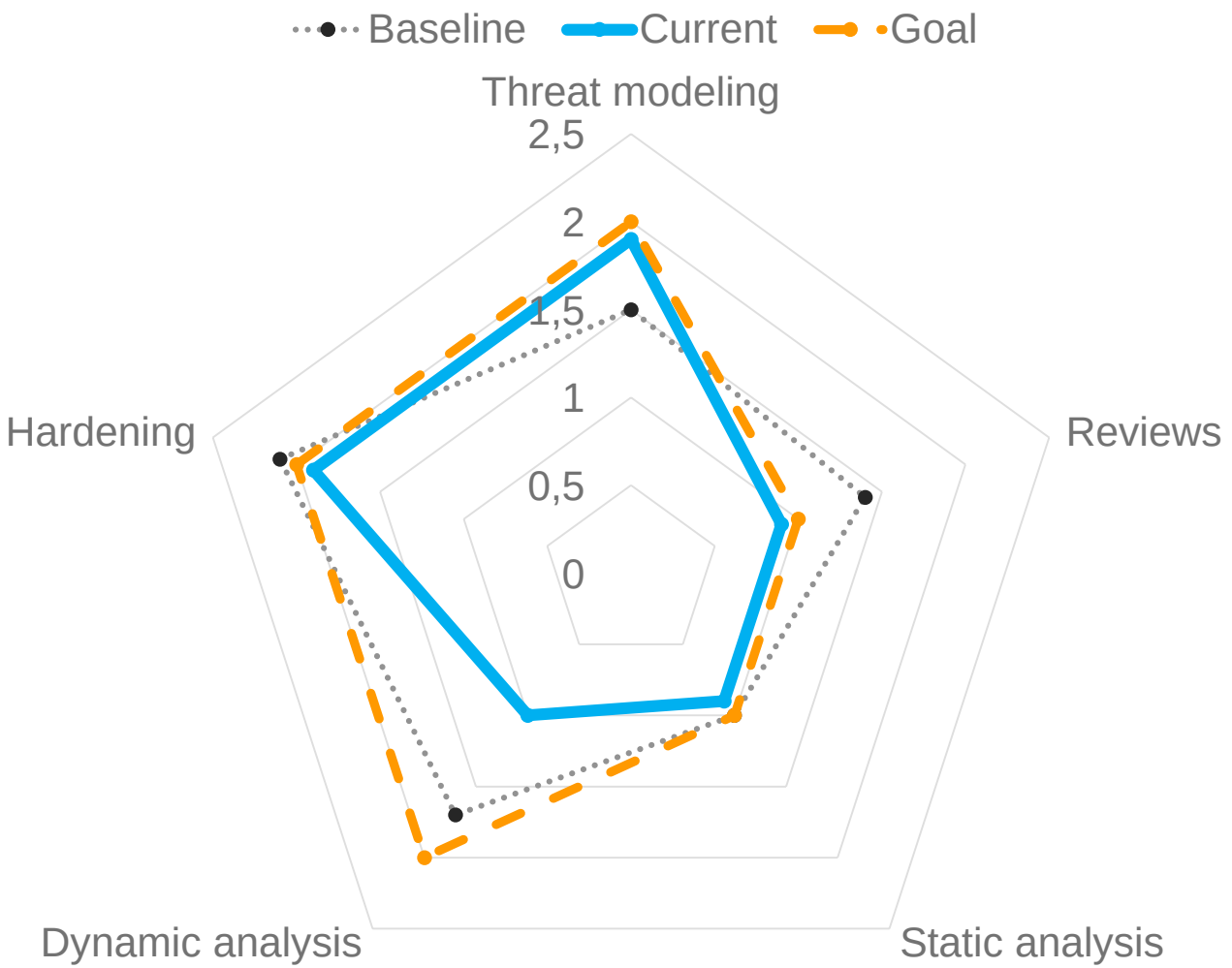
- **CR1: Opportunistically find basic code-level vulnerabilities**
- **CR2: Make code review more accurate through automation**
- **CR3: Mandate code review to discover language-level and app-specific risks**

YES/NO

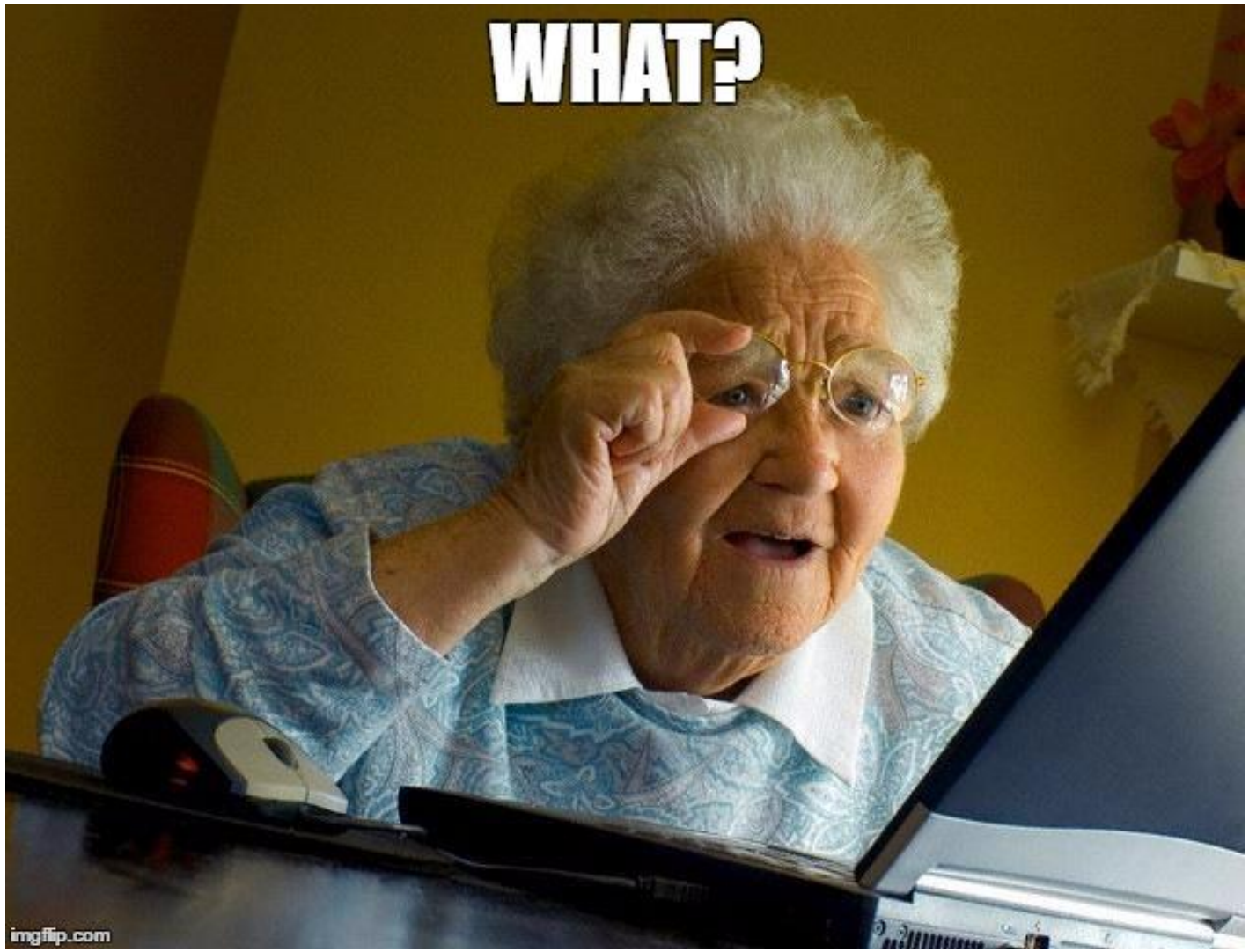
All components' code cleaned up based on static analysis at least quarterly?

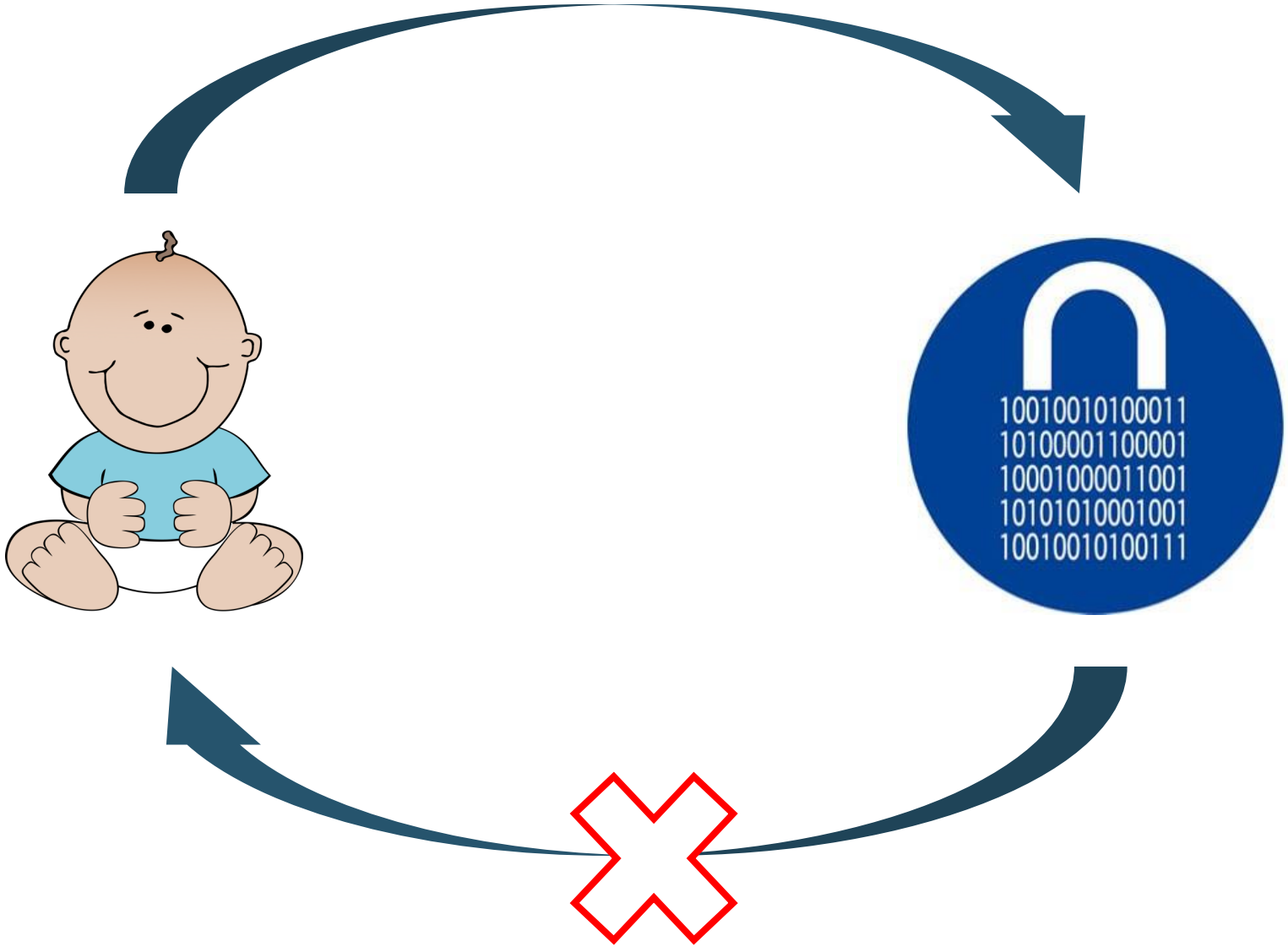
Static analysis breaks release candidate builds for all new code?

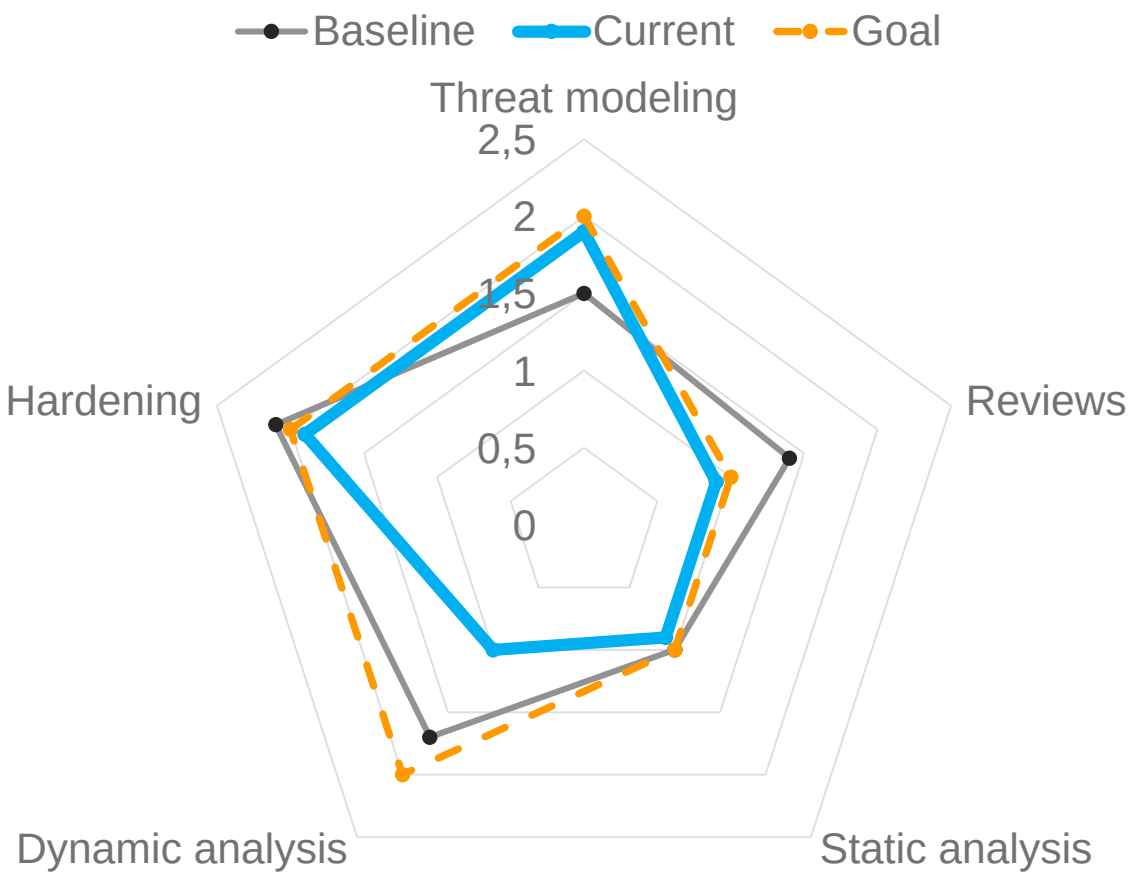
Static analysis running daily on all code with clean report?


Baseline
Current
Goal
Threat modeling
2,5
2
1,5
1
0,5
0
Reviews
Static analysis
Dynamic analysis
Hardening

nagyszülők | gross elders | Großeltern


WHAT?
imgflip.com

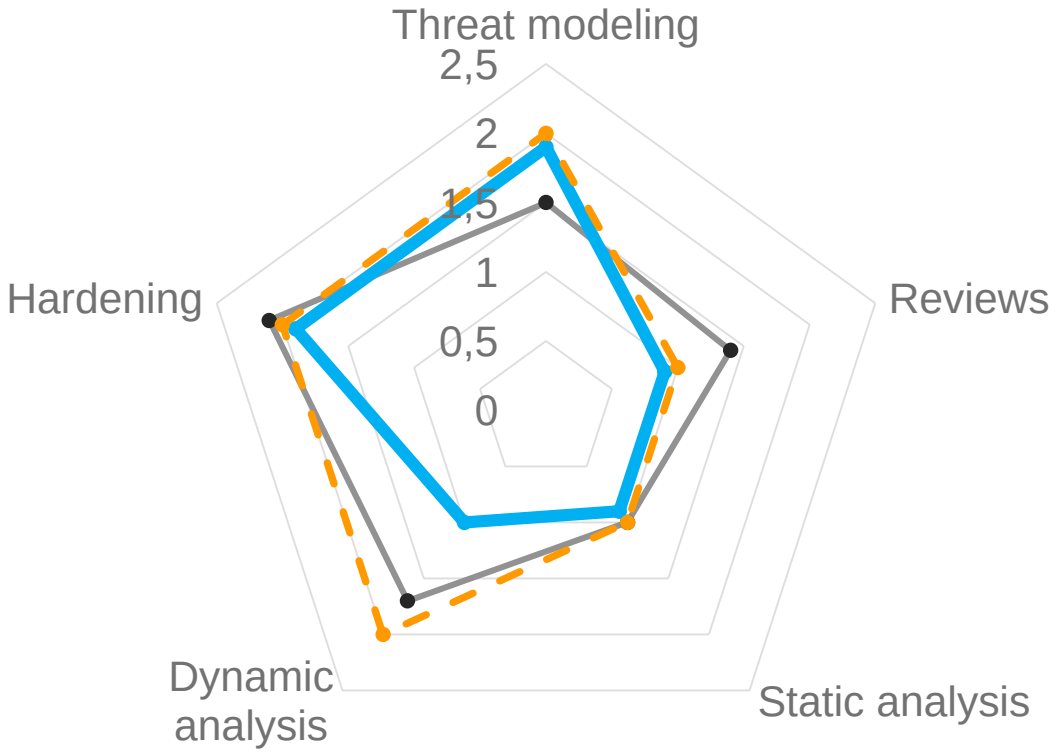

10010010100011
10100001100001
10001000011001
10101010001001
10010010100111

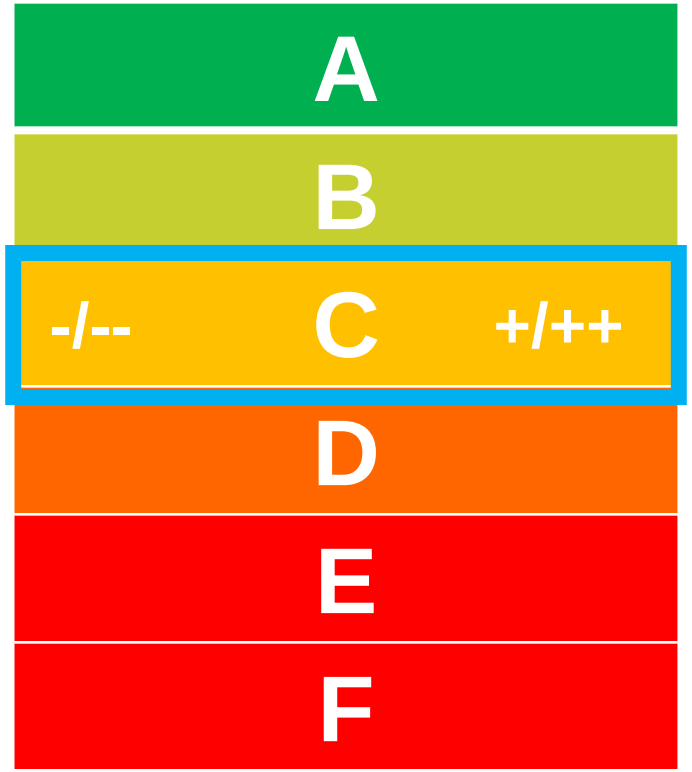


5 categories * 3 points = 15

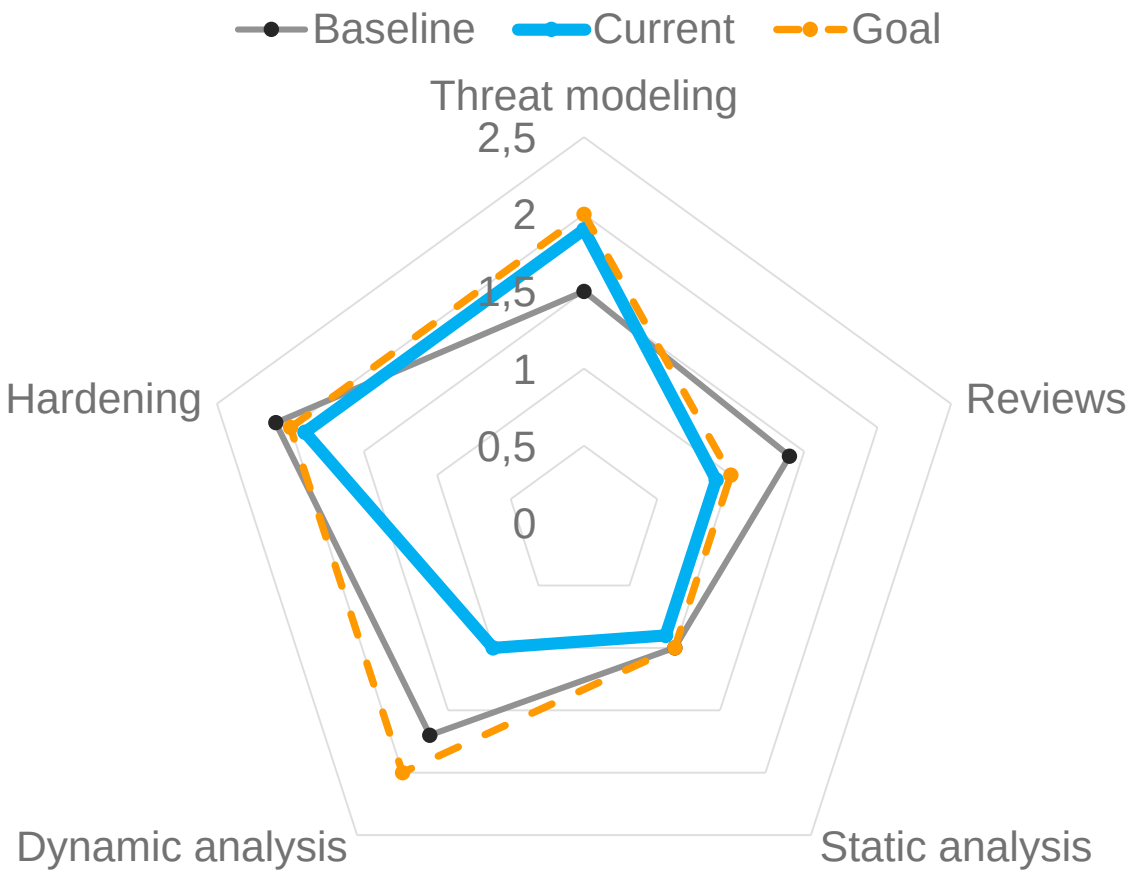
12-15	A
9-11	B
6-8	-/-- C +/++
4-5	D
2-3	E
0-1	F

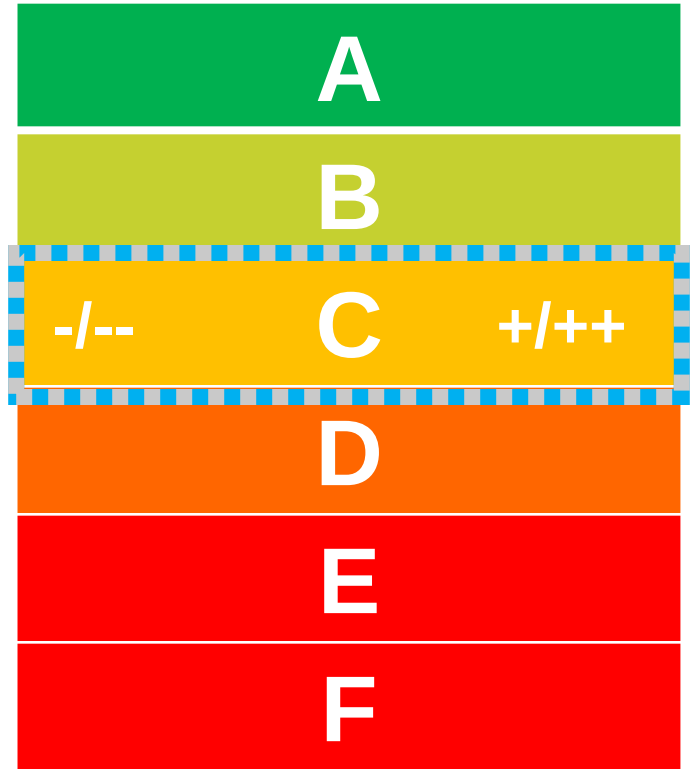
How do we stand?: *15 yes/no questions & +/-*



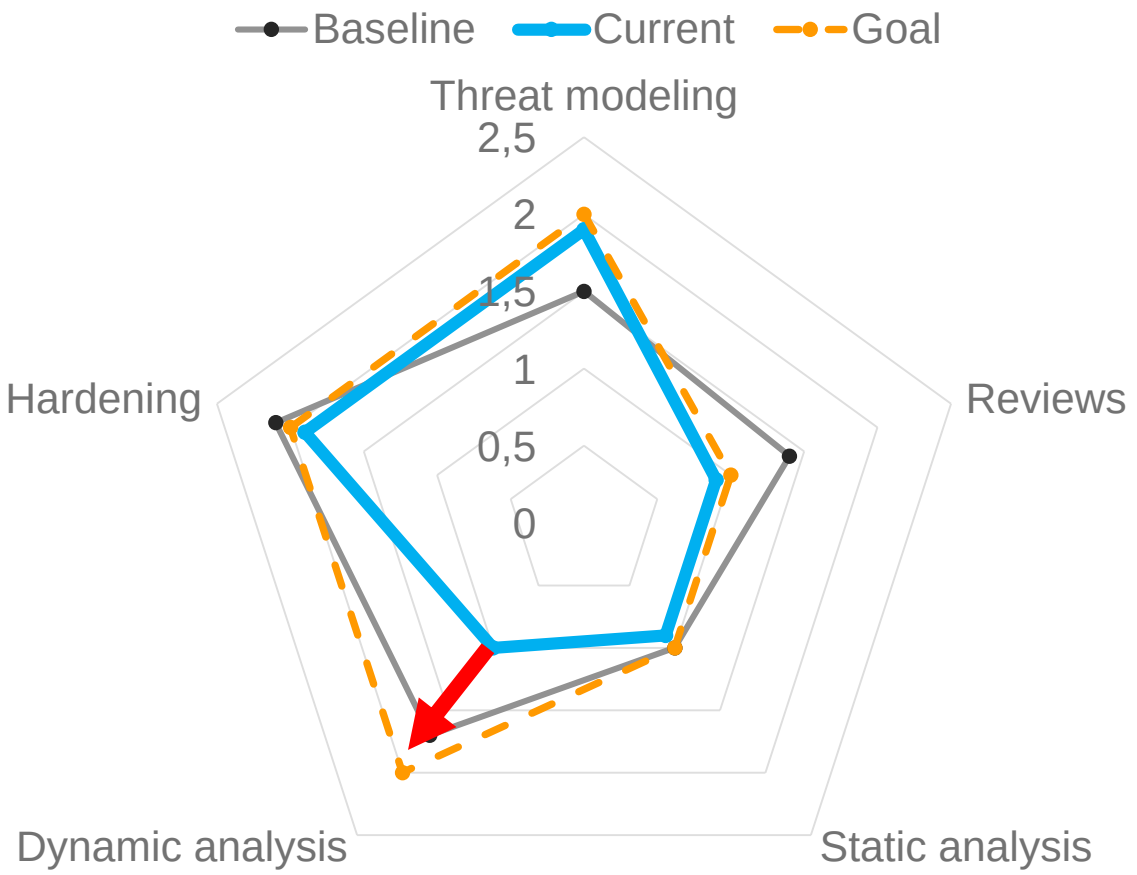


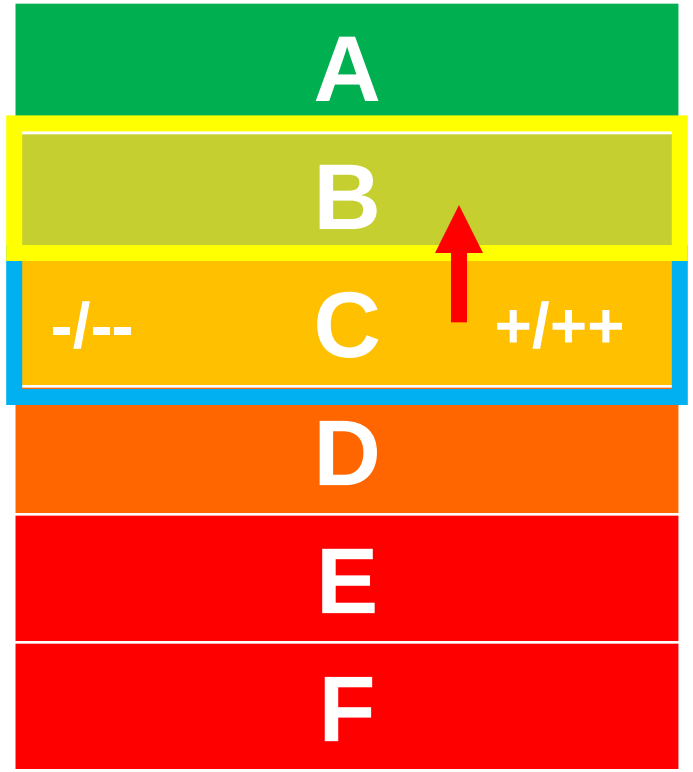
How do we do compared to others?: *baseline vs current*



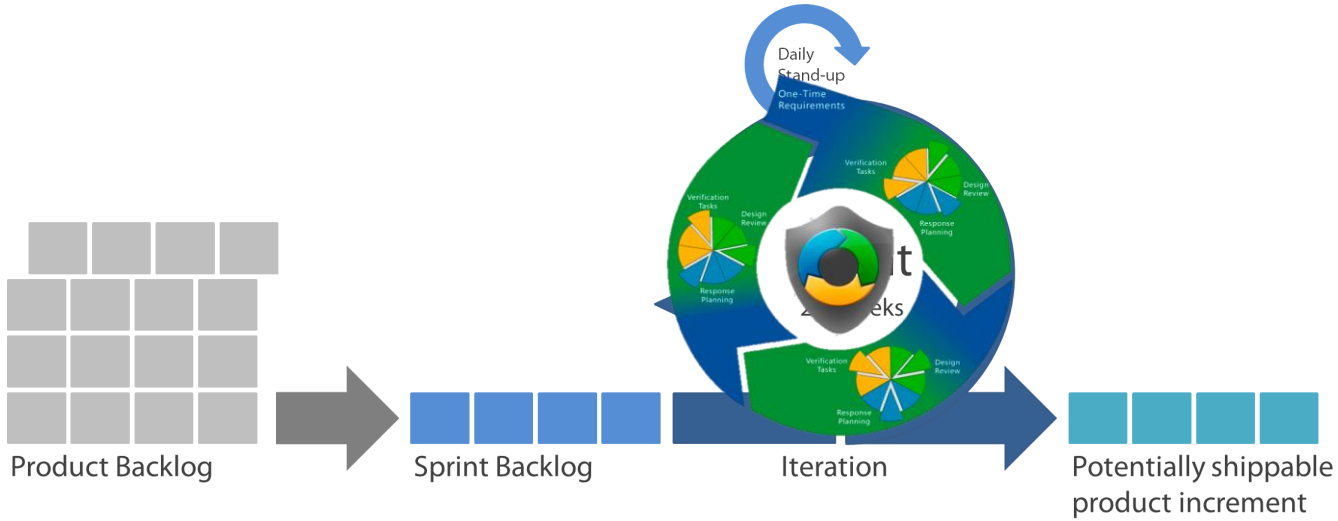


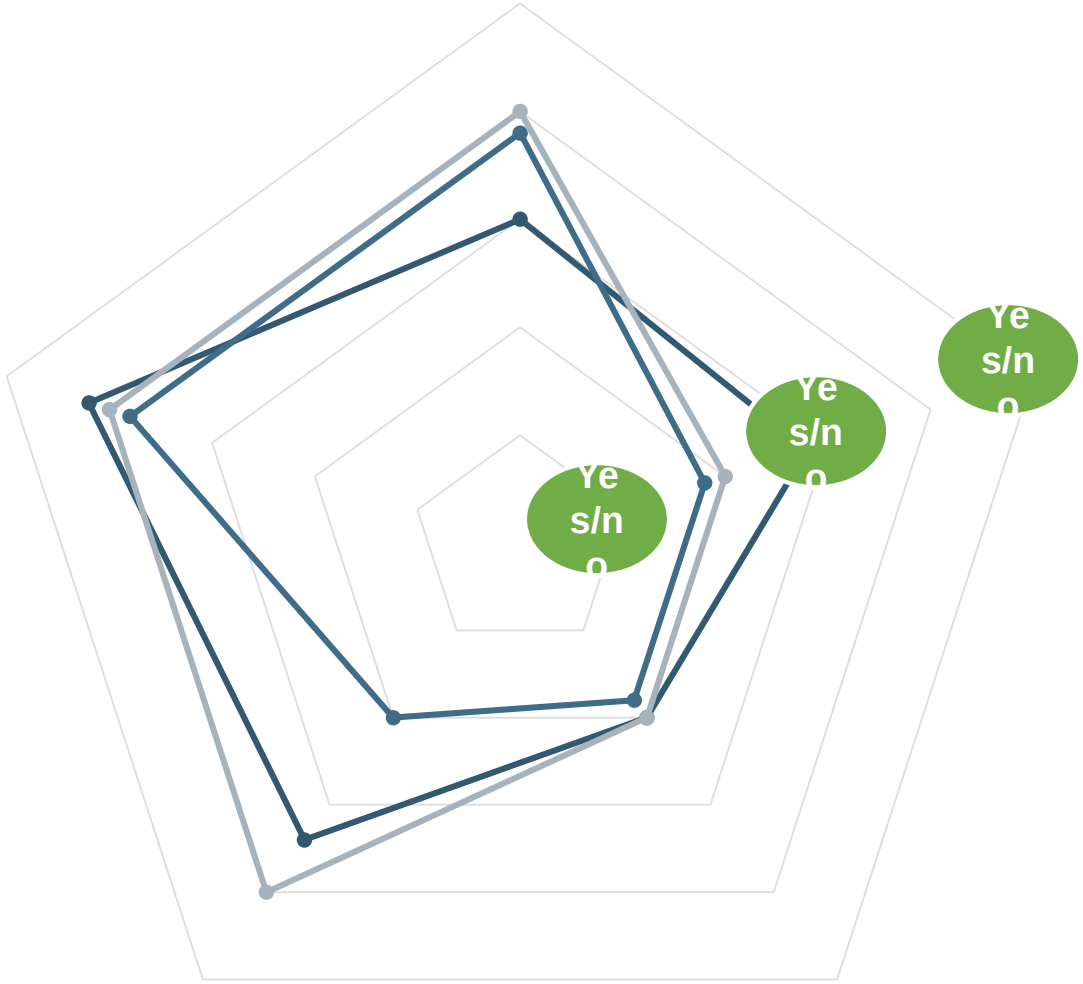
What is the goal for end of the year?: *detailed and high level planning*

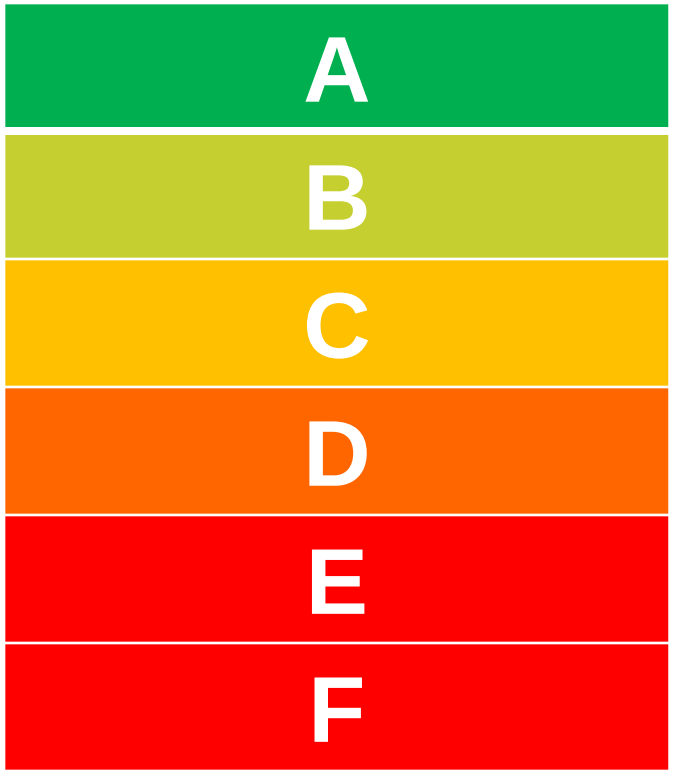





LEGO
LEGO
STAR WARS
5.99


Daily
Stand-up
One-Time
Requirements
Verification
Tasks
Design
Review
Response
Planning
Product Backlog
Sprint Backlog
Iteration
Potentially shippable
product increment


Yes/no
Yes/no
Yes/no



- ball

- LEGO

- teddy

Thank you.

LogMeIn