# SAMM Sessions - Generic DevOps Security Maturity Model by Timo Pagel – Notes

**Introduction**

Target: Optimisation of Time to Market

Old – Deploy once a year. DevOps deploys multiple times an hour

Problems when delivering fast:

People – tool developer wants to use every new framework to create software operations want the stability and want to keep things on track, no new framework.

Fast iterations, you want roles to be mixed to increase efficiency.

Quality assurance is important but it is vital that everyone know they are responsible for quality.

How to enhance information security? In devops strategies and through devops strategies

**DevOps dimensions**

Virtual environments - KVM/Docker: No network restrictions/Docker: No Patch Information.

Build and deployment process – Developer pushes code to version control, build and deployment that fetches libraries from external repos, then code moves to internal repo, then Production system or production near system.

Testing needs fast feedback, because vulnerabilities can cost a lot of money when found later. Integration tests, module tests, static analysis. Should also be applied on Test System. Prod near system Active tests such as load tests and vulnerability scans. Production system should have passive vulnerability scans and passive infrastructure tests. **Counter – Scans should be moved to an earlier step in the process.**

Information gathering – collection, processing and visualisation of system and application metrics. Alarming.

**Maturity model**

Culture and organisation infrastructure build and deployment,

test and verification dynamic depth static depth consolidation application test infrastructure test

information gathering monitoring and metric logging

Static depth – Different levels with dependencies between dimensions with a defined build process.

Level 2 static analysis for important server side components.

Implementation points covering – Risk, counter measure, impact for security / hardness of implementation, influence of information security triad dependencies, implementation hints

Assigning of implementation points GRAPH Y Axis Outcome (impact) for security VL to VH, X Axis Hardness of implementation VL to VH. Needed knowledge (1,2,3, 4, Disciplines) Needed time and needed resources (systems).

Test for server side application components with known vulnerabilities – Dimension: static depth. Risk: Server side components can have vulnerabilities. Outcome for security very high

**Conclusion**

Hard to determine the impact for security and the hardness of implementation. Technology changes can also influence metrics.

Points raised:

Granular security practices compared to SAMM.

Merging this model with SAMM would be quite difficult due to the detail -operation agnostic.

**Key Takeaways**

No need to change the core model but reviewing the SAMM activities is important to assure they are implementation neutral.

We can complement this with implementation advice for e.g. waterfall, agile and DevOps.

Evening session for devops security maturity model mapping to SAMM is scheduled

Devops security maturity model is a potential implementation subset of SAMM which is more abstract and implementation neutral.

The effort/impact dimensions make for an interesting addition to activities.