

## Unit 2

# UML Modelling to Support Secure System Planning

## Seminar 1 – Scrum Security Review

References to security activities are derived from four most common security engineering processes, namely, Cigatel Security Touchpoints (“CT”), Common Criteria (“CC”), Microsoft SDL (“MS”) and CLASP (“CL”).

## Engineering Process Definitions

- **Cigatel Security Touchpoints.** Lightweight integration of core activities in existing development processes to improve product security.

Activities include *Security Requirements, Abuse Cases, Risk Analyses, Assumption Documentation, Static Code Analyses, Penetration Testing, Red Team Testing, Risk Based Testing, External Reviews.* (Baca & Carlsson, 2011)

- **Common Criteria.** An ISO-certified process and users can specify their security and functional requirements.

Activities include *Security Requirements, Security Definitions, Risk Analyses, Critical/Vulnerable Assets, UMLSec, Requirements Validation, Repository Improvement.* (Baca & Carlsson, 2011)

- **Microsoft SDL.** Modified to suite agile environments and categorises activities based on *frequency of use.* (Sharma & Bawa, 2020).

Activities include *Security Requirements, Role Matrix, Design Requirements, Quality Gates, Cost analysis, Threat Modelling, Attack Surface Reduction, Security Tools, Coding Rules, Static Analysis, dynamic Analysis, Fuzzy Testing,*

Code Reviews, Incident Response Planning, Final Security Review. (Baca & Carlsson, 2011)

- **CLASP (Comprehensive, Lightweight Application Security Process).** Provided by OWASP and provides best practices, activities and roles to integrate security.

Activities include Awareness, Threat Modelling, Resource and Trust Boundaries, Security Analysis of Requirements, Security Metrics, Specify Operational Environment, Global Security Policy, User Roles and Capabilities, Identify Attack Surface, Security Design Principles, Security Architecture, Code Signing, Security Tests, Source-level Security Review, Operational Planning and Readiness.

## Proposed Mapping

Table 1 SCRUM mapping to security activities

SCRUM Lifecycle	Recommended Security Activity
Vision	<ul style="list-style-type: none"> <li>▪ Design requirements [MS] <i>Ensure design specifications <u>fulfil security requirements</u>.</i></li> </ul>
Majority Activity: <b>Security Requirements</b>	<ul style="list-style-type: none"> <li>▪ <b>Security requirements</b> [CG, CC, MS] <i>Identification of <u>security and privacy concerns</u> handled by security experts.</i></li> <li>▪ Security definitions [CC] <i><u>Common language</u> across the organisation of security-related terms and policies.</i></li> <li>▪ Education [CL] <i>All stakeholders are <u>informed</u> about importance of security engineering and highlights possible breaches.</i></li> </ul>

SCRUM Lifecycle	Recommended Security Activity
<b>Product Backlog</b>  Majority Activity: <b>Resource protection</b>	<ul style="list-style-type: none"> <li>▪ Role matrix [MS, CL] <i>Identification of <u>roles</u> and their <u>access levels</u>. Important for authentication and authorization.</i></li> <li>▪ Security architecture</li> <li>▪ Acceptance criteria</li> <li>▪ <b>Identify resources and trust boundaries [CL]; Critical/Vulnerable Assets [CC]; Attack Surface Reduction [MS];</b> <i>Structures <u>network access</u> and interaction points.</i></li> </ul>
<b>Sprint Planning</b>  Majority Activity: <b>Risk Analysis</b>	<ul style="list-style-type: none"> <li>▪ <b>Risk analyses</b> [CG, CC] <i>Validates areifacts and quality of work and <u>mitigates existing architecture flaws</u>.</i></li> <li>▪ Coding rules [MS] <i><u>Insecure functions</u> are swapped out for safer alternatives.</i></li> </ul>
<b>Sprints</b>  Majority Activity: <b>Static Code Analysis</b>	<ul style="list-style-type: none"> <li>▪ <b>Static code analysis</b> [CG, MS] <i><u>Source code</u> analysed for <u>vulnerabilities</u>.</i></li> <li>▪ Security testing [CL]</li> <li>▪ Code reviews [MS, CL]</li> </ul>
<b>Sprint Review</b>  Majority Activity: <b>Testing</b>	<ul style="list-style-type: none"> <li>▪ Dynamic analysis [MS] <i>Tools to aid in detecting <u>memory corruption, privilege concerns and other security issues</u>. Enables mitigation in following sprints.</i></li> </ul>

SCRUM Lifecycle	Recommended Security Activity
	<ul style="list-style-type: none"> <li>▪ <b>Vulnerability and penetration testing [CG]</b> <i><u>Simulates real-world conditions and attacks.</u></i></li> </ul>
Sprint Retrospective	<ul style="list-style-type: none"> <li>▪ Final security review [MS]</li> <li>▪ Incident response planning [MS] <i>Provides <u>direction</u> on actions required in a security emergency.</i></li> </ul>
Majority Activity: <b>Risk Analysis</b>	

## Conclusion

In line with the findings from Sharma & Bawa (2020), the Microsoft SDL and CL indeed do play a vital role in incorporating security concerns into a SCRUM process (as shown in Table 1).

## References

- Baca, D. & Carlsson, B., (2011) Agile development with security engineering activities. *Proceedings of the 2011 International Conference on Software and Systems Process*:149-158.
- Sharma, A. & Bawa, R.K. (2020) Identification and integration of security activities for secure agile development. *International Journal of Information Technology*: 1-14.

## Bibliography

- Moyón, F., Almeida, P., Riofrío, D., Mendez, D. & Kalinowski, M., (2020). Security Compliance in Agile Software Development: A Systematic Mapping Study. *46th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*: 413-420.