### **Ensuring Secure Software in the Scrum Agile Life Cycle**

The Scrum agile life cycle is a widely adopted approach to software development that emphasizes iterative progress through collaborative efforts and adaptive planning. While Scrum effectively enhances productivity and responsiveness to changing requirements, integrating security practices into each stage is crucial to producing secure software. Based on the principles from (Sharma & Bawa2020), this document outlines the key stages of the Scrum agile life cycle and recommends security processes to ensure that security is an integral part of the development process.

### **Table: Scrum Agile Development Stages and Recommended Security Processes**

|  |  |
| --- | --- |
| **Scrum Agile Development Stages** | **Recommended Security Processes** |
| **Product Backlog Creation** | - **Threat Modeling**: Identify potential security threats and vulnerabilities for each backlog item.  - **Security Requirements Definition**: Include security requirements and acceptance criteria for each feature or user story. |
| **Sprint Planning** | - **Security Risk Assessment**: Assess and prioritize security risks for the planned sprint items.  - **Security Task Allocation**: Allocate specific security tasks (e.g., code reviews, security testing) within the sprint backlog. |
| **Sprint Development** | - **Secure Coding Practices**: Implement secure coding guidelines and standards (e.g., OWASP guidelines).  - **Automated Security Testing**: Integrate automated security testing tools (e.g., static analysis, dependency checking) into the CI/CD pipeline.  - **Pair Programming and Code Reviews**: Encourage collaborative coding and peer reviews focusing on security. |
| **Daily Scrum** | - **Security Stand-Up**: Include a brief discussion on any security issues or updates during the daily scrum.  - **Continuous Monitoring**: Monitor any security alerts or issues in real-time as development progresses. |
| **Sprint Review** | - **Security Demonstration**: Demonstrate and verify security features and compliance during the sprint review.  - **Feedback Integration**: Collect and integrate feedback on security from stakeholders and product owners. |
| **Sprint Retrospective** | - **Security Retrospective**: Review security practices and incidents from the completed sprint to identify areas for improvement.  - **Process Improvement**: Update security processes and practices based on lessons learned. |
| **Release** | - **Final Security Assessment**: Conduct a thorough security assessment, including penetration testing, before the final release.  - **Documentation and Training**: Provide comprehensive security documentation and training for end-users and maintenance teams.  - **Deployment Monitoring**: Ensure security measures are in place for post-deployment monitoring and incident response. |

### **Conclusion**

Integrating security processes into each stage of the Scrum agile life cycle is essential for developing secure software. By adopting practices such as threat modeling, secure coding, automated security testing, and continuous monitoring, development teams can proactively address security risks and ensure robust protection against vulnerabilities. This approach not only enhances the overall security posture of the software but also fosters a culture of security awareness and accountability within the team. By following these recommendations, organizations can achieve a balance between agility and security, delivering high-quality, secure software products.

**References**

Sharma, A., & Bawa, R. K. (2020). Identification and Integration of Security Activities for Secure Agile Development. *International Journal of Information Technology*.