- Remember that security is a process and not a product
- This can be done by leveraging dynamic application security testing (DAST), static application security testing (SAST), and software composition analysis (SCA) or dependency tracking tools during standard automated release workflows or on a regularly scheduled basis
- Good security testing requires going beyond what is expected and thinking like an attacker who is trying to break the application
- automated tools do not think creatively.
- It is critical not to perform a superficial security review of an application and consider it complete.
- It is important to track the results of testing engagements, and develop metrics that will reveal the application security trends
- Writing the report should not be overly burdensome on the security tester themselves. Security testers are not generally renowned for their creative writing skills
    - Decomposing the application – use a process of manual inspection to understand how the application works, its assets, functionality, and connectivity.
    - Defining and classifying the assets – classify the assets into tangible and intangible assets and rank them according to business importance.
    - Exploring potential vulnerabilities - whether technical, operational, or managerial.
    - Exploring potential threats – develop a realistic view of potential attack vectors from an attacker's perspective by using threat scenarios or attack trees.
    - Creating mitigation strategies – develop mitigating controls for each of the threats deemed to be realistic.

4. The following figure shows a typical proportional representation overlaid onto the SDLC. In keeping with research and experience, it is essential that companies place a higher emphasis on the early stages of development.

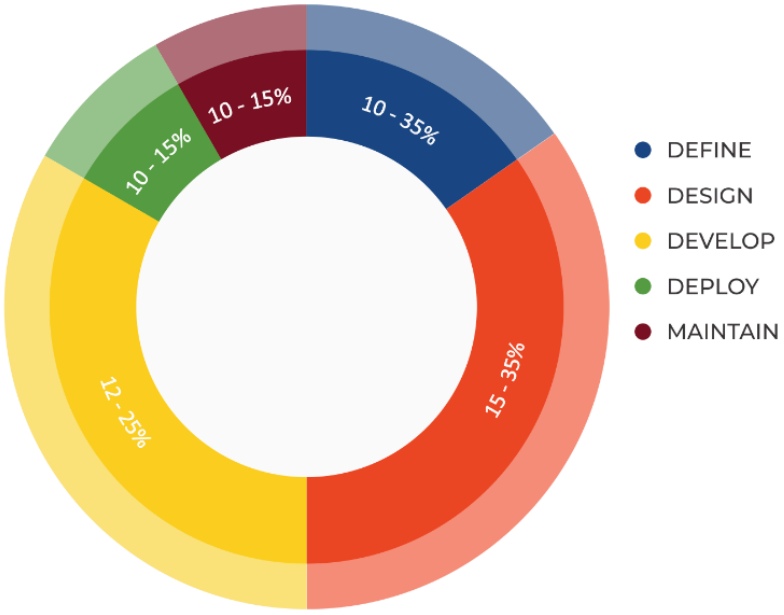

Figure 2-3: Proportion of Test Effort in SDLC

The following figure shows a typical proportional representation overlaid onto testing techniques.

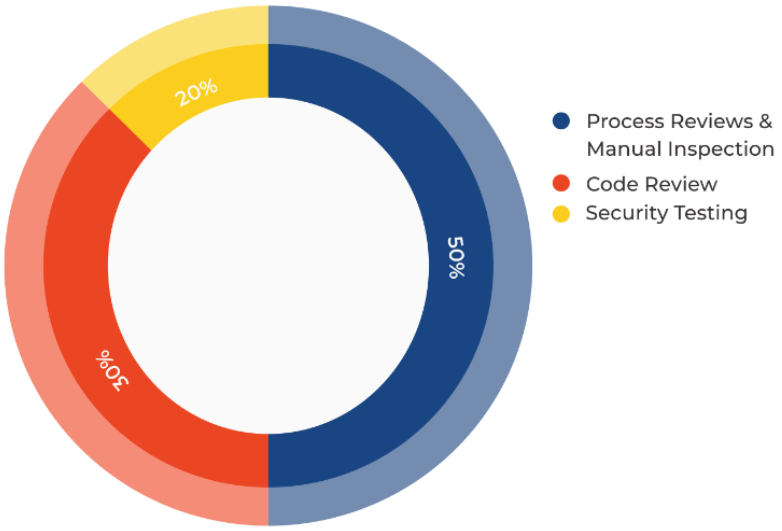

Figure 2-4: Proportion of Test Effort According to Test Technique

-
- A general checklist of the applicable regulations, standards, and policies is a good preliminary security compliance analysis for web applications
- Another section of the checklist needs to enforce general requirements for compliance with the organization's information security standards and policies.
    - From the security assessment perspective, security requirements can be validated at different phases of the SDLC by using different

artifacts and testing methodologies. For example, threat modeling focuses on identifying security flaws during design; secure code analysis and reviews focus on identifying security issues in source code during development; and penetration testing focuses on identifying vulnerabilities in the application during testing or validation.

- The first evidence of a potential SQL injection vulnerability that can be validated is the generation of a SQL exception. A further validation of the SQL vulnerability might involve manually injecting attack vectors to modify the grammar of the SQL query for an information disclosure exploit. This might involve a lot of trial-and-error analysis before the malicious query is executed. Assuming the tester has the source code, they might directly learn from the source code analysis how to construct the SQL attack vector that will successfully exploit the vulnerability
- Penetration Testing Execution Standard (PTES) defines penetration testing as 7 phases:
    - Pre-engagement Interactions
    - Intelligence Gathering
    - Threat Modeling
    - Vulnerability Analysis
    - Exploitation
    - Post Exploitation
    - Reporting
- The major area of penetration testing includes:
    - Network Footprinting (Reconnaissance)
    - Discovery & Probing
    - Enumeration
    - Password cracking
    - Vulnerability Assessment
    - AS/400 Auditing
    - Bluetooth Specific Testing
    - Cisco Specific Testing
    - Citrix Specific Testing
    - Network Backbone

    - Server Specific Tests\
    - VoIP Security
    - Wireless Penetration
    - Physical Security
    - Final Report - template
-