PACIFIC NW SOFTWARE QUALITY CONFERENCE

Building Security into Your App One Story at a Time

BHUSHAN GUPTA
OCTOBER 11, 2022

About Me


Vegetarian


TEA!
Sit, Sip, &
Forget About
the World
Tea
TEA

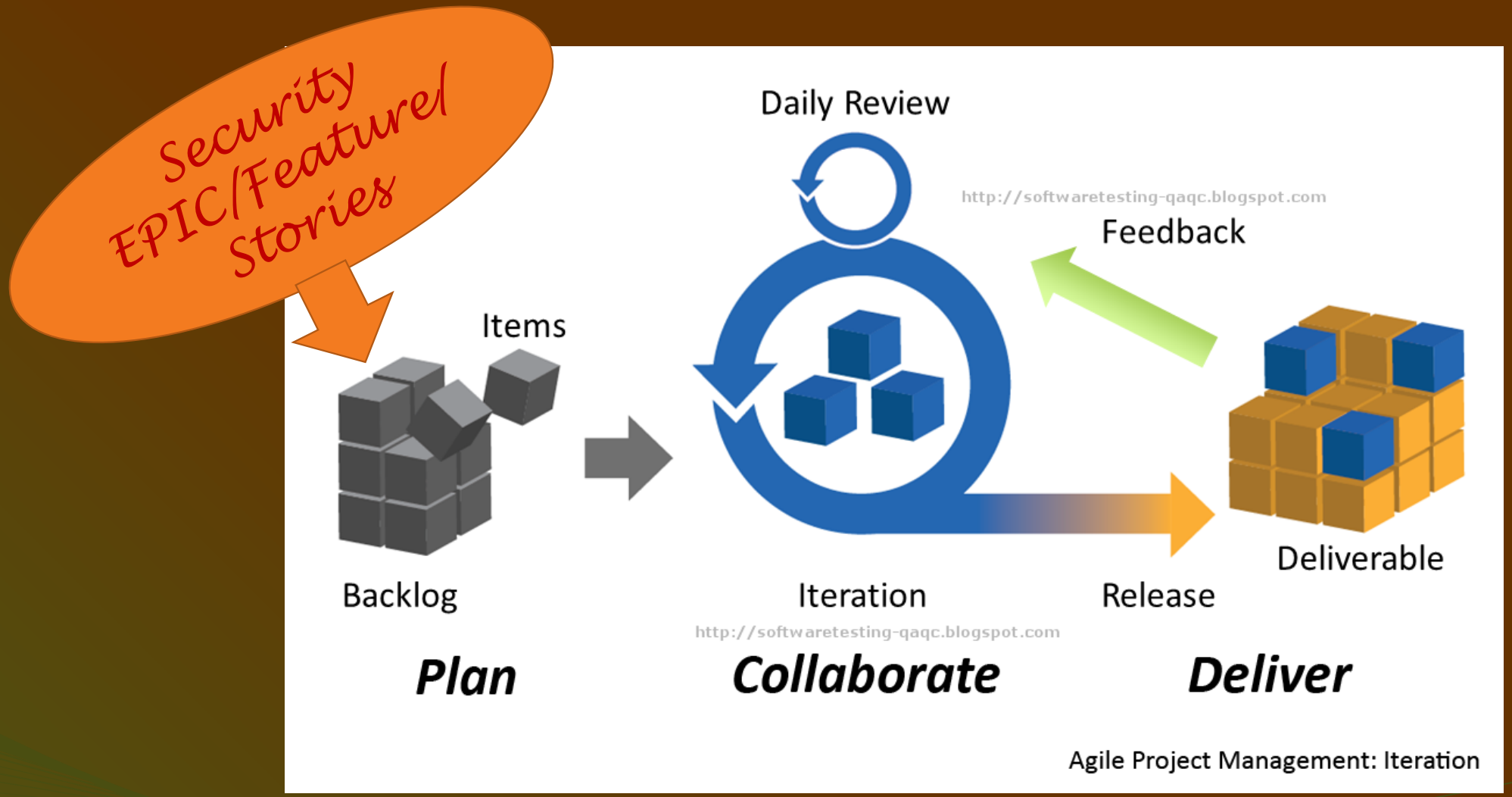

IMAGE
envision
.com


Do
Plan
Check
Quality
Quality
Champion

Agenda

- Agile SDLC framework with Security Component
- Security Pillars and Controls
- Stakeholders/Actors and RACI
- 3 Ws – What, When, Who
- Conclusion

Agile SDLC Framework



Are all stories equally vulnerable?

STRIDE

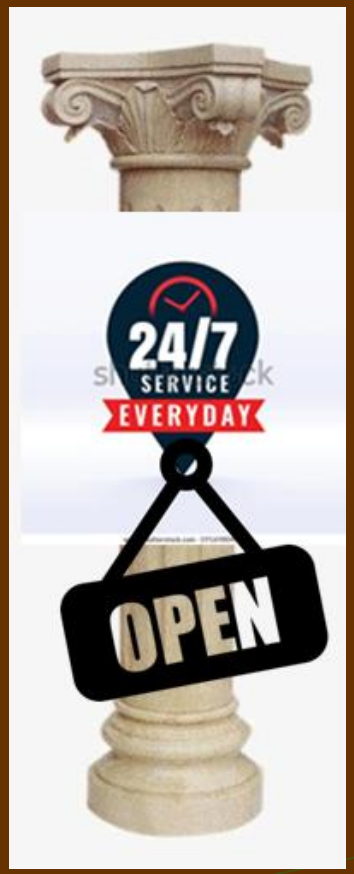
What is STRIDE?

Category	Description
Spoofing	Gaining Access to the system using false identity
Tampering	Unauthorized modification of data
Repudiation	Ability to successfully deny an activity already taken place
Information Disclosure	Unwanted exposure of private data
Denial of Service	Making system unavailable for use
Elevation of Privilege	Assuming identity of a privileged user from limited privileges

Three Pillars of Security

Availability

Security Controls



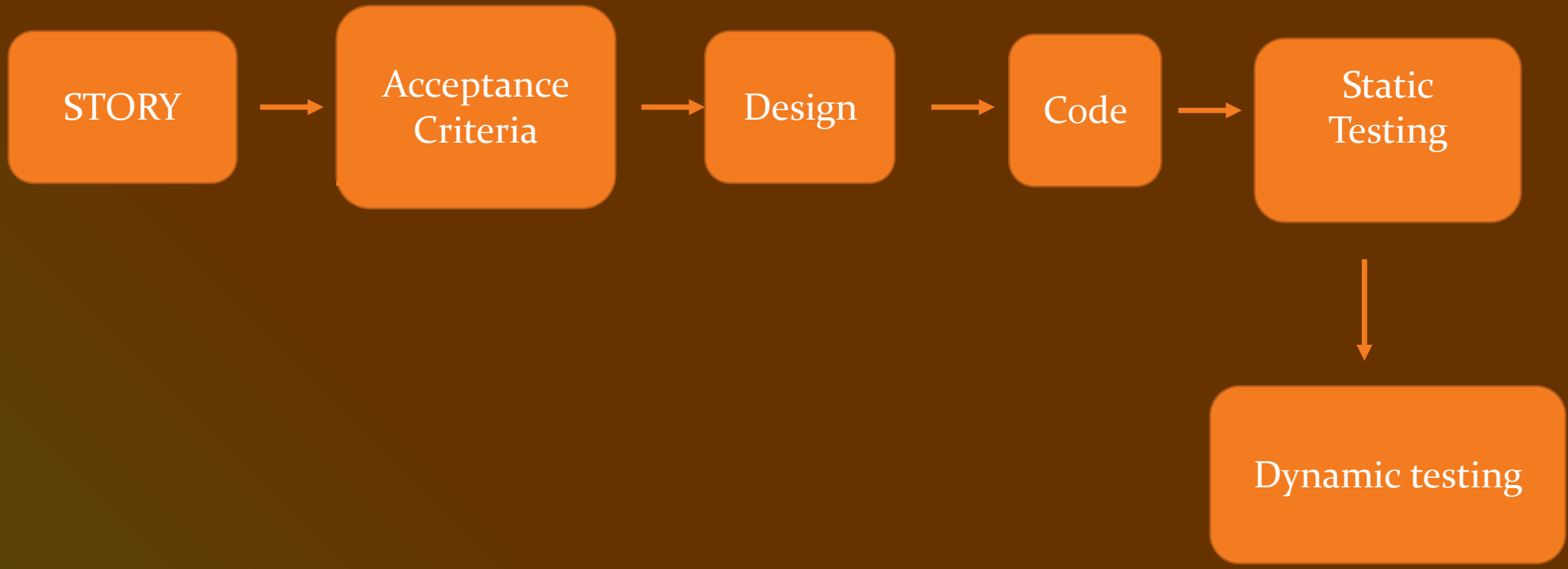
Security Controls – Story 6, Login

Spoofing	Validate client data, Use of prepared statements
Tampering	• Secure Network Protocol, Data Encryption • Update logs
Repudiation	Not applicable
Information Disclosure	Hide password while entering Impalement Session Securing Methods
Denial of Service	Resolve any handshake problems between the client and the server
Elevation of Privilege	Validate client data, Use of prepared statements

Security Control – How do they work??

- Policies
- Processes
- Guidelines

Story Progression



RACI Framework

- R - Responsible
- A - Accountable
- C - Consulted
- I - Informed

Actors:

Product Owner
Development Engineer
Security Engineer
QA Engineer
Management Team

Story Context – Sign Up

Criteria ID	Acceptance Criteria
01	Collect only essential PII from the user
02	Hide sensitive data at submission (login ID, password)
03	Protect user data exposure when in motion
04	Protect user data exposure when stored
05	Protect user data from unauthorized changes when in motion and stored

RACI

R	A	C	I
Product Owner	Scrum Master	Security Engineer	Need based

From Acceptance Criteria to Security Controls

ID	Description	Security Control
01	Collect only essential PII from the user	None/security policy
02	Hide sensitive data at submission (login ID, password)	Mask LoginID and Password at the data entry
03	Protect user data exposure when in motion	Encryption and Secure Transmission
04	Protect user data exposure when stored	Encryption
05	Protect user data from unauthorized changes when stored	Prevent privilege escalation Implement lowest level of privilege

R	A	C	I
Security Engineer	Scrum Master	Product Owner	Need based

Design/Development Considerations

Best Practice	Example
Defense in depth	Two-factor authentication
Fail secure	System lockup if an unsafe activity is suspected
Protect weakest code	
Principle of least privilege	Allow lowest level of privilege
Use of prepared statement	Building database queries

R	A	C	I
Development Lead Engineer	Development Manager	Product Owner, Security Engineer	Need based

Static Testing

- When: Code is RAW or compiled
- What: Code review, Compilation Errors
- Who:

R	A	C	I
Development Engineer (s)	Development Lead Engineer	Product Owner, Security Engineer	Need based

Dynamic Testing

- When: Code is executable
- What: Scan the code using scanners
- Who:

R	A	C	I
Test Engineer(s)	Test Manager	Product Owner, Security Engineer, Development Engineer	Need based

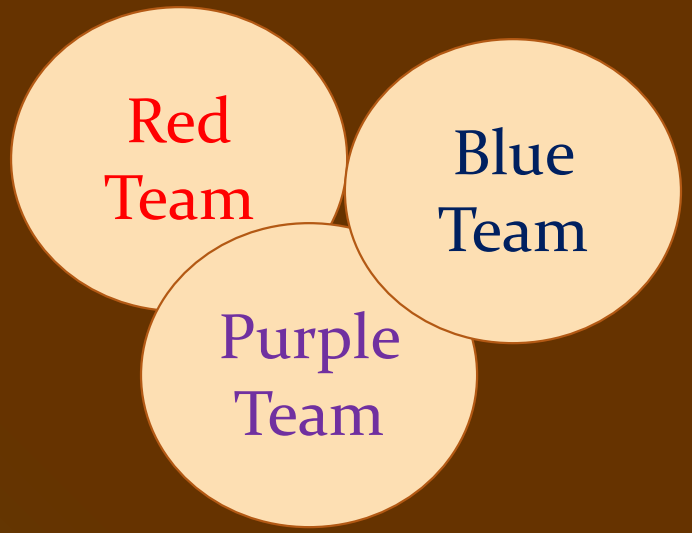
Testing Priorities

Factors:

- Risk Assessment/Vulnerability Level
- Developer confidence
- Known Breaches

R	A	C	I
Test Engineer(s)	Test Manager	Product Owner, Security Engineer, Development Engineer	Need based

Testing Teams



Conclusion

- Education and awareness
- Organizational planning
- Ownership at each stage of story development
- Collaboration between teams

Acceptance Criteria – Story 6, Login

Spoofing	Client unable to enter malicious information (injection attack)
Tampering	• Data secure while in motion with compatible browsers • Login event recorded for auditing (both successful & Unsuccessful)
Repudiation	Not applicable
Information Disclosure	Social engineering such as shoulder surfing is not possible Secure Session (No MitM attack)
Denial of Service	Does not cause network overload
Elevation of Privilege	Client unable to enter malicious information (injection attack)

Security Controls – Story 6, Login

Spoofing	Validate client data, Use of prepared statements
Tampering	• Secure Network Protocol, Data Encryption • Update logs
Repudiation	Not applicable
Information Disclosure	Hide password while entering Impalement Session Securing Methods
Denial of Service	Resolve any handshake problems between the client and the server
Elevation of Privilege	Validate client data, Use of prepared statements

Code Reviews/Inspections

Targets:

- Development Environment
  - Opsys and Language shortcomings – Buffer Overflow
- Process and User Access Rights, (Admin vs. normal user)
- Input Validation – both on client and server side
- Coding Practices - Use of prepared statements vs. dynamic queries
- Runtime Environment Verification – System Hardening

Environment Hardening

Reinforce your environment:

- Configuration management policy
  - Versioning – apply updates immediately
  - Change Control
- Login level – Admin vs user
- Principle of Least privilege
- Change Default Settings – File path, access rights, password
- Disabling unnecessary features