Security Strategies in Web Applications and Social Networking

Lesson 11

Testing and Quality Assurance for Production Web Sites

Learning Objective

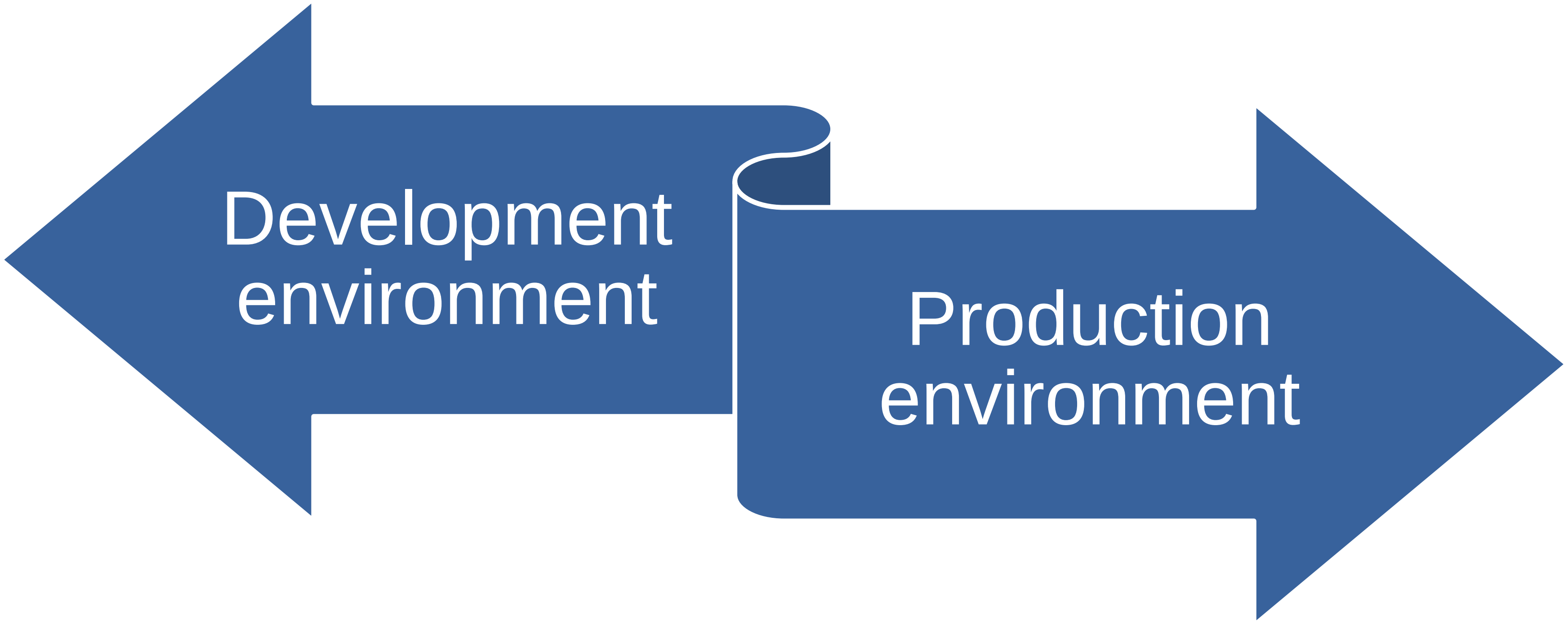
- Analyze the role and importance of quality assurance (QA) testing for Web applications.

Key Concepts

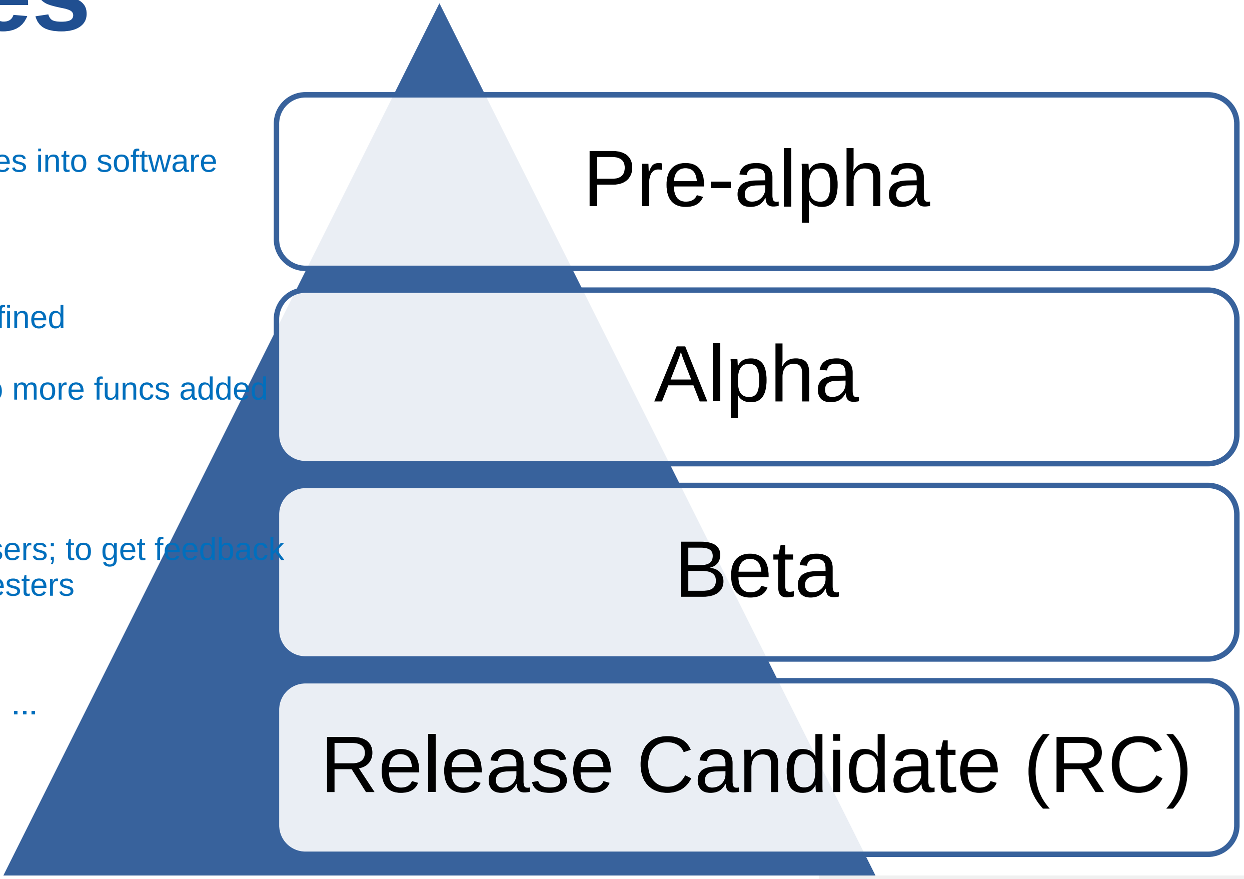
- Configuration and change management
- Quality assurance testing and gap analysis
- Metrics and measurement programs
- Monitoring production applications
- Strategies and best practices

Development vs. Production Environments

controlled vs uncontrolled env
ideally fix all bugs before moving into prod env



Software Development Life Cycle Stages

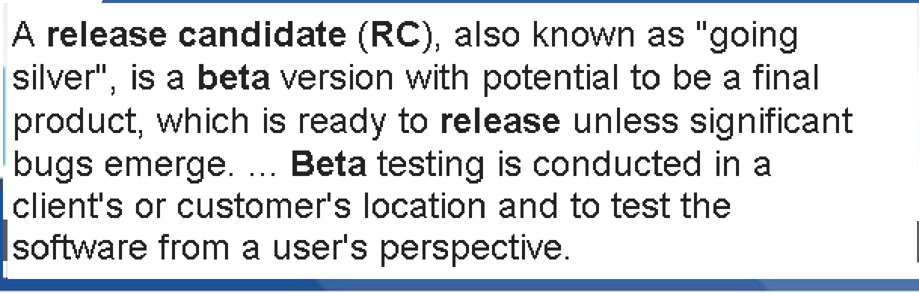


intro stage
devs still including features into software

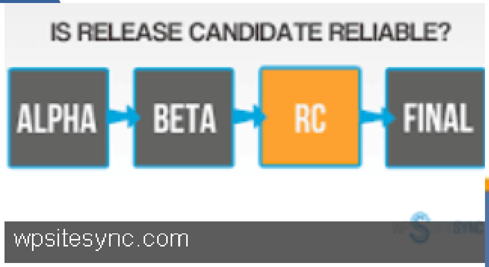
apps are coded, funcs defined
bugs are found and fixed
code/feature freeze => no more funcs added

limited release version
given access to limited users; to get feedback
users become software testers

can have multiple: rc1, rc2, ...
until confident to release



A **release candidate** (**RC**), also known as "going silver", is a **beta** version with potential to be a final product, which is ready to **release** unless significant bugs emerge. ... **Beta** testing is conducted in a client's or customer's location and to test the software from a user's perspective.



Formalizing Software and System Changes

- Change management
  - Provides all stakeholders in the application the knowledge of *what* will be changed and *when* it will be changed
- A formal deployment plan is a best practice for separation of duties from developers, QA analysts, and system administrators.

Formalizing Software and System Changes (Continued)

- Configuration management
  - Provides system administrators and developers with documented history of changes in case a rollback to a previous version or full restore of an application is necessary

Network Documentation

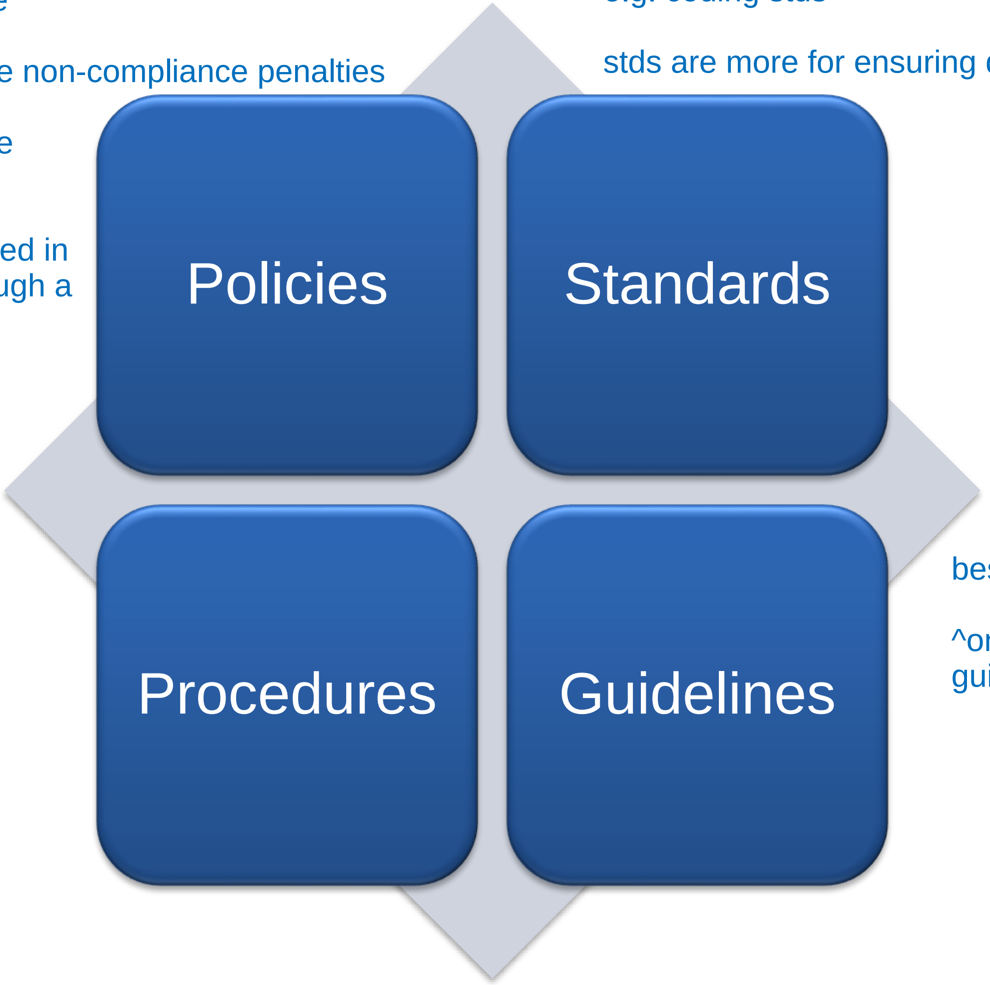
org wide documented rule on how things shuld be done, who does it, and what is done
e.g. email policies, etc.
enrforcable because there are non-compliance penalties

diff from regulations which are punishable by law

anything desired to be enforced in an organization must go through a policy

means of ensuring quality
e.g. coding stds

stds are more for ensuring quality/performance



step by step things

best practices; not enforceable

^only diff b/t policies and guidelines

Web Site Deployment Checklist

- Verify links https://validator.w3.org/checklink tests redirects, anchors, etc.
- Test browser compatibility
- Test all downloads
- Verify digital certificates and Secure Sockets Layer (SSL) URLs work correctly
- Test forms and form controls
- Verify path traversal
- Review navigational structure
- Verify shopping features cc transactions, database, inventory, etc.
- Web page load times performance, stress testing

Software Testing Techniques

not concerned with code but testing func/non-func features

interested in testing the code too

combination of both...?

- Black box testing
- White box testing
- Gray box testing
- Unit testing
- Integration testing
- System testing
- Regression testing
- Usability testing

how user friendly, intuitive UI
how easy to use

Software Testing Techniques

load testing -> under normal load conditions, check the performance
stress testing -> putting env under extreme stress, check how system responds

e.g. if sys shuts off or sth happens.
how long does it take for sys to come back up

Performance testing

Software stress testing

Recovery testing

Security testing

Compatibility testing

Regulatory compliance testing

diff browsers
also compatibility with other 3rd party components

e.g. PCIS compliance, etc.

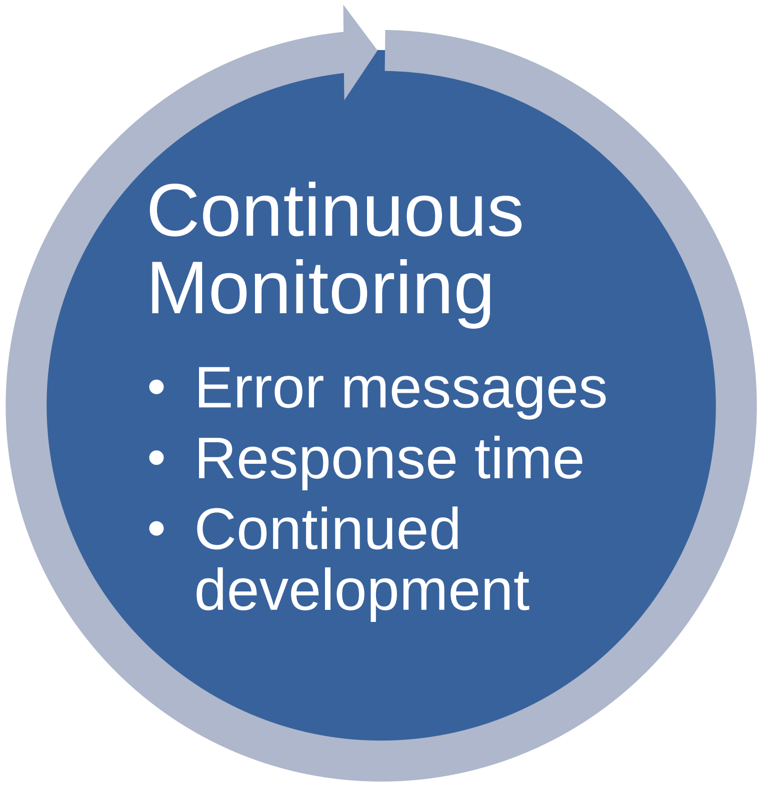
Key Areas of Security Vulnerability Testing

- Application design
- Default security measures
  - set to lowest privileges
  - only enable the ones used
- Mass deployment security
- Information and response abilities

Mitigating Security Holes

- Outline vulnerabilities
- Classify vulnerabilities
- Develop a mitigation plan
- Retest

Production Deployment



Continuous Monitoring

- Error messages
- Response time
- Continued development

Analyzing Web Page Statistics

Analytics	Browser statistics
	Bounce rate
	Network performance
	Visitor paths
	Shopping cart abandonment
	Visitor location

Best Practices

- Protect data
- Minimize data collection
- Use tracking software
- Conduct usability tests
- Ongoing security testing
- Develop standards, policies, and procedures
- Use regression testing
- Use a testing cycle

Summary

- Configuration and change management
- QA test plans
- Metrics and measurement programs
- Monitoring production applications
- Formalizing software and system changes

Virtual Lab

- Performing Dynamic and Static Quality Control Testing

If your educational institution included the Jones & Bartlett labs as part of the course curriculum, use this script to introduce the lab:

"In this lesson, you learned about quality assurance testing for Web applications. You explored configuration and change management, gap analysis, metrics and measurement programs, monitoring production applications, and strategies and best practices.

In the lab for this lesson, you will use skipfish, a dynamic testing tool, to identify vulnerabilities in the Damn Vulnerable Web Application (DVWA). The DVWA is a Web application that is made purposefully vulnerable. It is installed on a local Web server to allow security analysts a safe place to test the security of their applications. You also will use RATS (Rough Auditing Tool for Security) to perform static analysis testing on the DVWA. You will use the vi Editor to review the source code for a part of the DVWA to identify exactly where the software code is most vulnerable. Finally, you will compare the results of both skipfish and RATS reports."