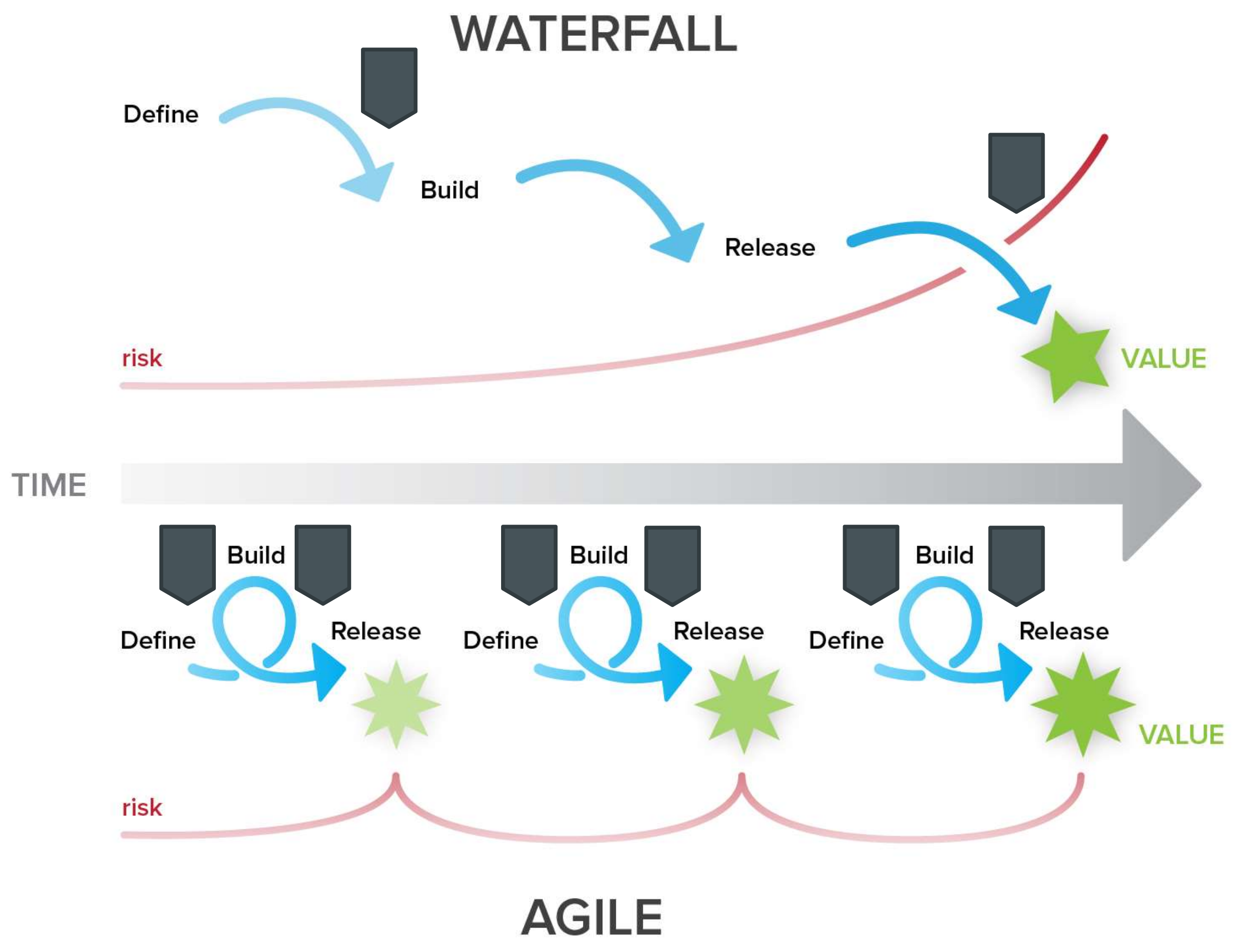
CASE STUDY - A JOURNEY TOWARDS DEVSECOPS

ANNIKA VATSA, M.B.A., SCRUM ALLIANCE CERTIFIED SCRUM PROFESSIONAL - SCRUM MASTER & PRODUCT OWNER
MANAGER, APPLICATION SECURITY, IDENTITY AND ACCESS MANAGEMENT AT CHOICE HOTELS INTERNATIONAL



Security Checkpoint ⬟

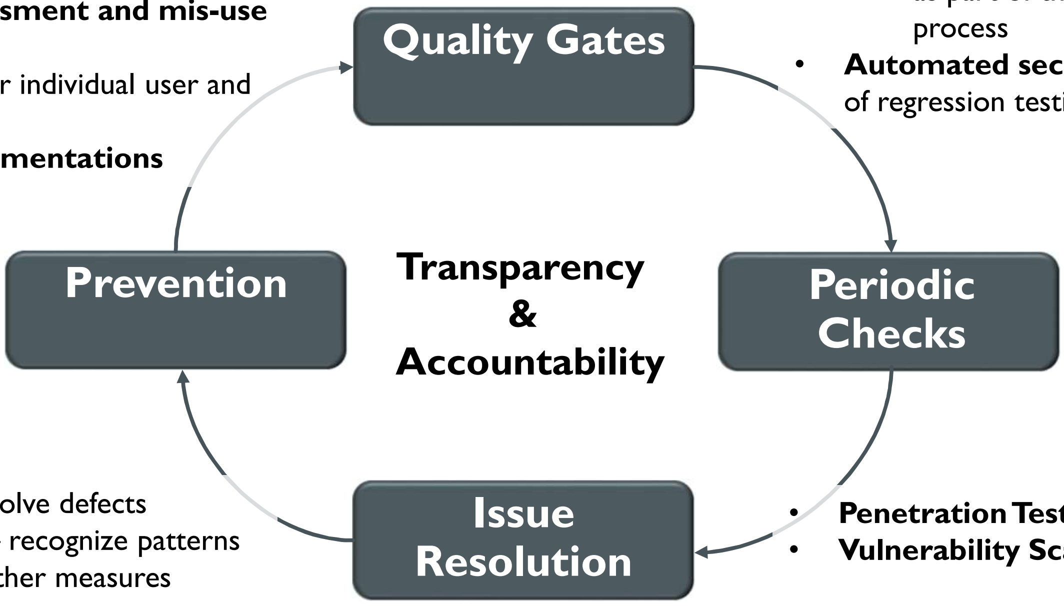
"[..] an increasing amount of [security testing]responsibility is being assigned to crossfunctional teams (across dev/ops/sec), and directly to developers—especially in faster organizations." SANS whitepapers 2017-state-application-security-balancing-speed-risk

Diagram Modified from Axian, Inc : Waterfall vs. Agile methodologies

SECURE SOFTWARE DEVELOPMENT VISION

- **Training**: security awareness, secure coding and other
- Follow Company Secure Coding Guidelines
- Leverage company's input validation library
- **Static Code Analysis** Tool embedded in IDE
- Security guidance from **assessments**
- **Feature based risk assessment and mis-use cases definitions**
- Set **access** appropriately for individual user and application accounts
- Leverage **reference implementations**

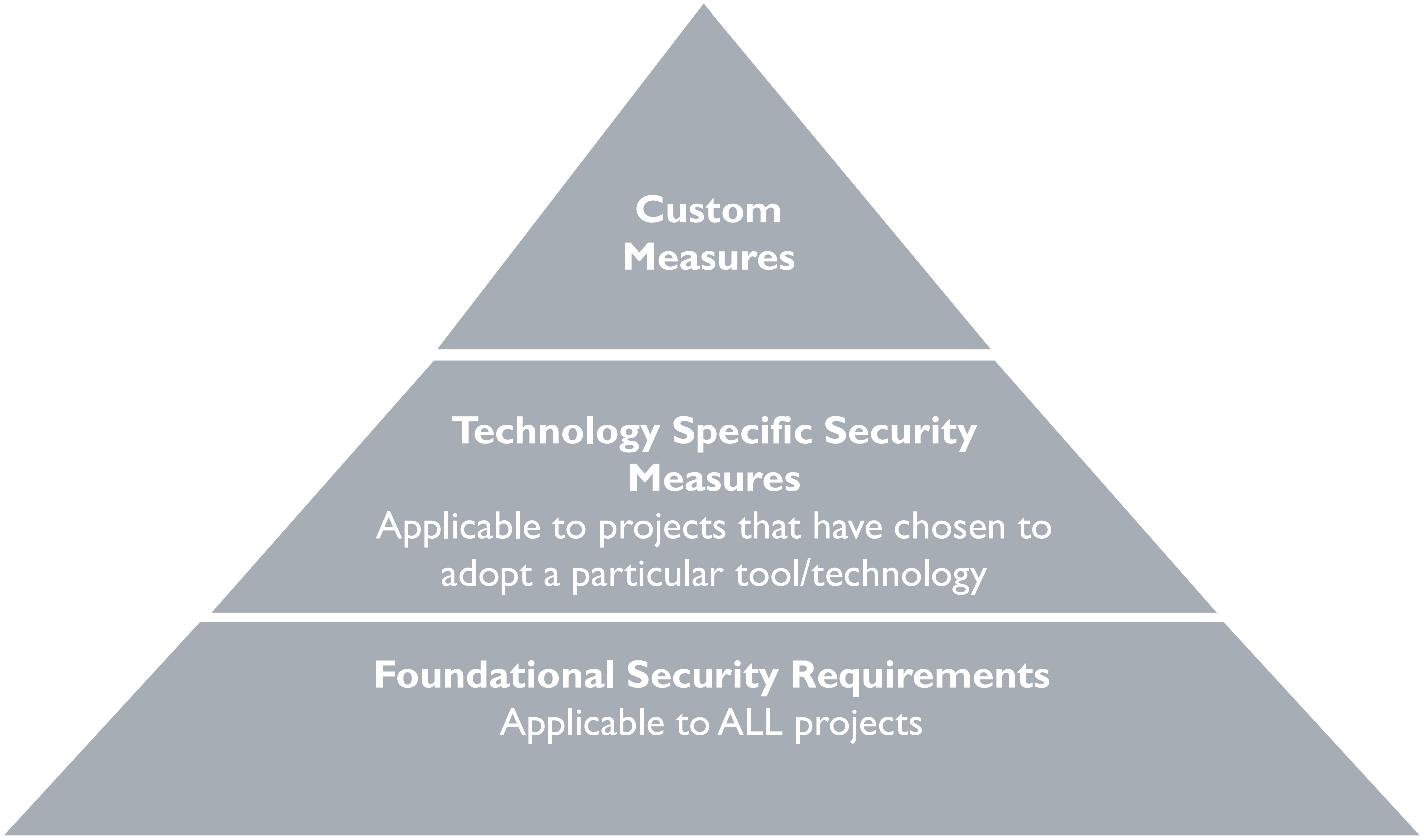
- **Static Code Analysis** feedback
  - in pull requests
  - as part of the automated build process
- **Automated security testing** as part of regression testing



- **Prioritize**, track and resolve defects
- **Root Cause Analysis** – recognize patterns and adjust training and other measures accordingly

- **Penetration Testing** (mostly manual)
- **Vulnerability Scans**

LAYERED APPROACH


Custom Measures
Technology Specific Security Measures
Applicable to projects that have chosen to adopt a particular tool/technology
Foundational Security Requirements
Applicable to ALL projects

TEMPLATES FOR USER STORIES

Feature Risk Assessment

- As Technology Risk Manager I can be assured that the features to be created are assessed for the risk they pose to the organization so that security measures prescribed can be balanced with that risk.
- Acceptance Criteria
  - Completed Feature Risk Assessment
  - Defined Mis-Use Scenarios for all High Risk Features

    Ask: "As a user what **can** I do with …"

    Rather than: "What **should** I do…"

Infrastructure Hardening

- As Technology Risk Manager I can be assured that the new production infrastructure set up in support of the project meets company security best practices so that the risk of a successful attack on the server is reduced to an acceptable level.
- Acceptance Criteria
  - Pass Vulnerability Scans
  - Setup File Integrity Monitoring
  - Setup Host-based Intrusion Detection

TEMPLATES FOR USER STORIES - CONTINUED

Secure Coding

- Code Review
- Static Code Analysis
- Penetration Testing

Logging

- Log important events according to standards
- Document what is being logged, where, log retention
- Integrate logs into SIEM per guidance from infosec team

Documentation & Business Continuity

- Network diagrams
- Service Accounts
- Data Classification
- Other Business Continuity Documentation such as 3rd party contacts, SLAs

STATIC CODE ANALYSIS – STEPPING INTO AUTOMATION

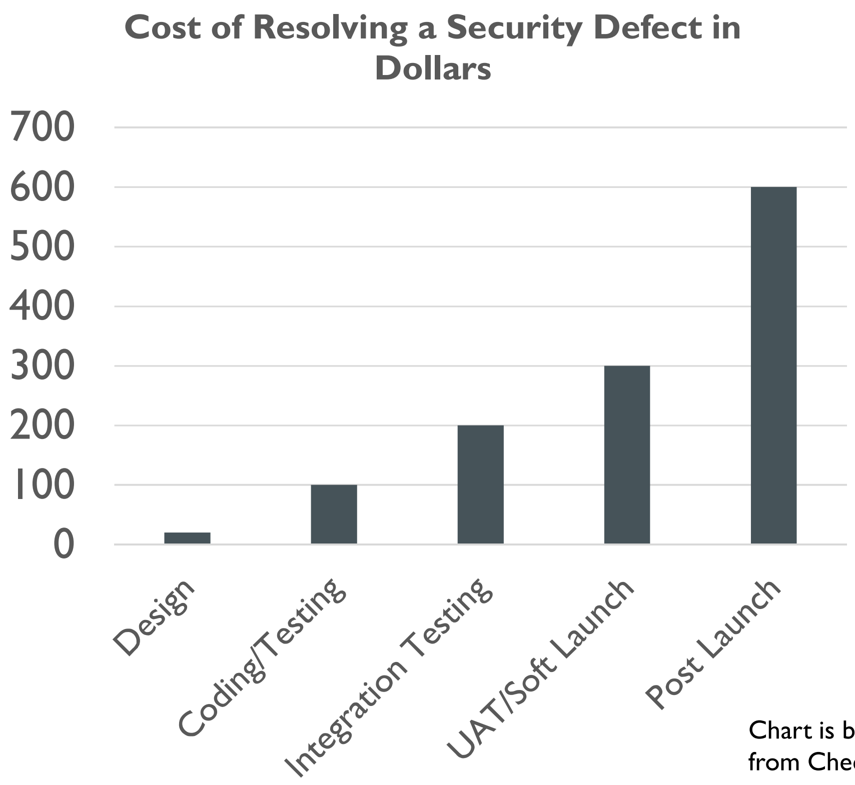


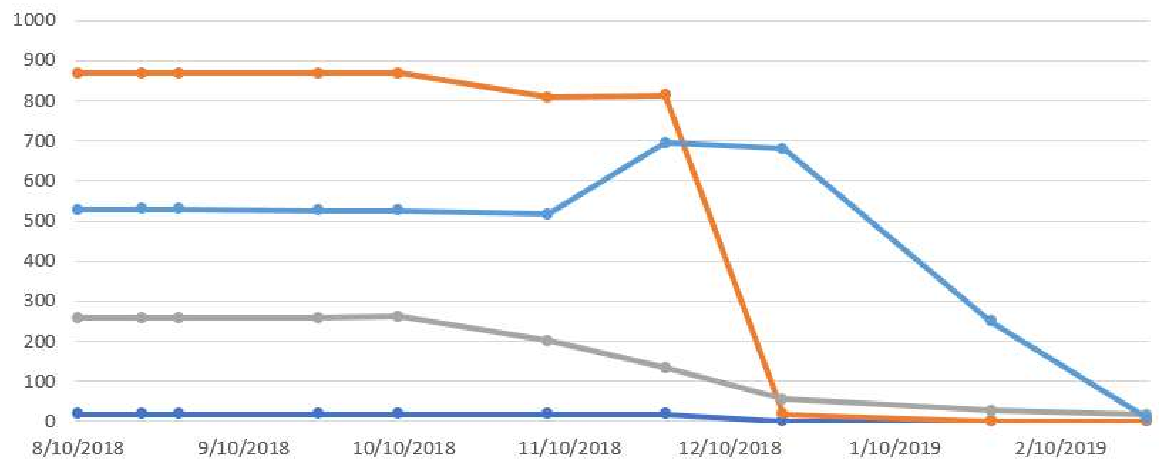
Chart is based on data from Checkmarx

- Selection Criteria
  - Compatibility with existing infrastructure
  - Total cost of ownership
  - Product Quality
    - Accuracy
    - Ease of use
    - Time/Effort Required for Scan
    - Issue Validation and Resolution Guidance
    - Issue Tracking & Reporting
  - Multiple integration options
  - Scales well

STATIC CODE ANALYSIS TOOL ROLLOUT

- Scheduled Scans
  - Initial scan configurations
  - InfoSec reviewed findings
  - Establish targets
- On-demand Scans
  - Developers become familiar with the tool
  - Manage towards targets
- Pipeline Integration
  - Ready for continuous checks
  - Maintain targets

Issue Counts for Various Application Groups



SECURITY RISK INDEX – FROM DATA TO DECISIONS

- $Risk\ Tolerance\ Utilization\ = \frac{sum\ of\ all\ weighted\ issues}{risk\ tolerance\ score\ of\ the\ application}$ **+** Δ

Example:

- $60\ \% = \frac{(5\ static\ code\ analysis\ issues\ times\ .8)+(2\ issues\ from\ penetration\ testing\ times\ 1)}{public\ data\ 5 + high\ availability\ 0 + no\ compliance\ requirements\ 5 + high\ visibility\ 0}$
- Δ **may be used to adjust to special business needs or to re-baseline if current utilization exceeds tolerance without that risk having been accepted officially**

Disclaimer: Numbers used in the example are for illustration purposes only.

CHANGE MANAGEMENT – EVALUATING RELEASE CANDIDATES

- Consistency & Transparency - Give a security score to every potential release.
- Avoid strict pass/fail - Acknowledge that exceptions may be necessary and plan on managing and tracking them.
- Couple loosely - Build a flexible system that will work even when underlying tools are switched out or new tools are added.

KEY TAKEAWAYS

- Prioritize
- Standardize
- Automate

- It’s an evolution not a destination. Continue to adjust and evolve!
- Allow for flexibility while driving towards standards.

Questions?