RSA CONFERENCE 2014
ASIA PACIFIC & JAPAN

Share.
Learn.
Secure.
Capitalizing on
Collective Intelligence

Continuous Delivery and Risk Management

SESSION ID: SEC-T10

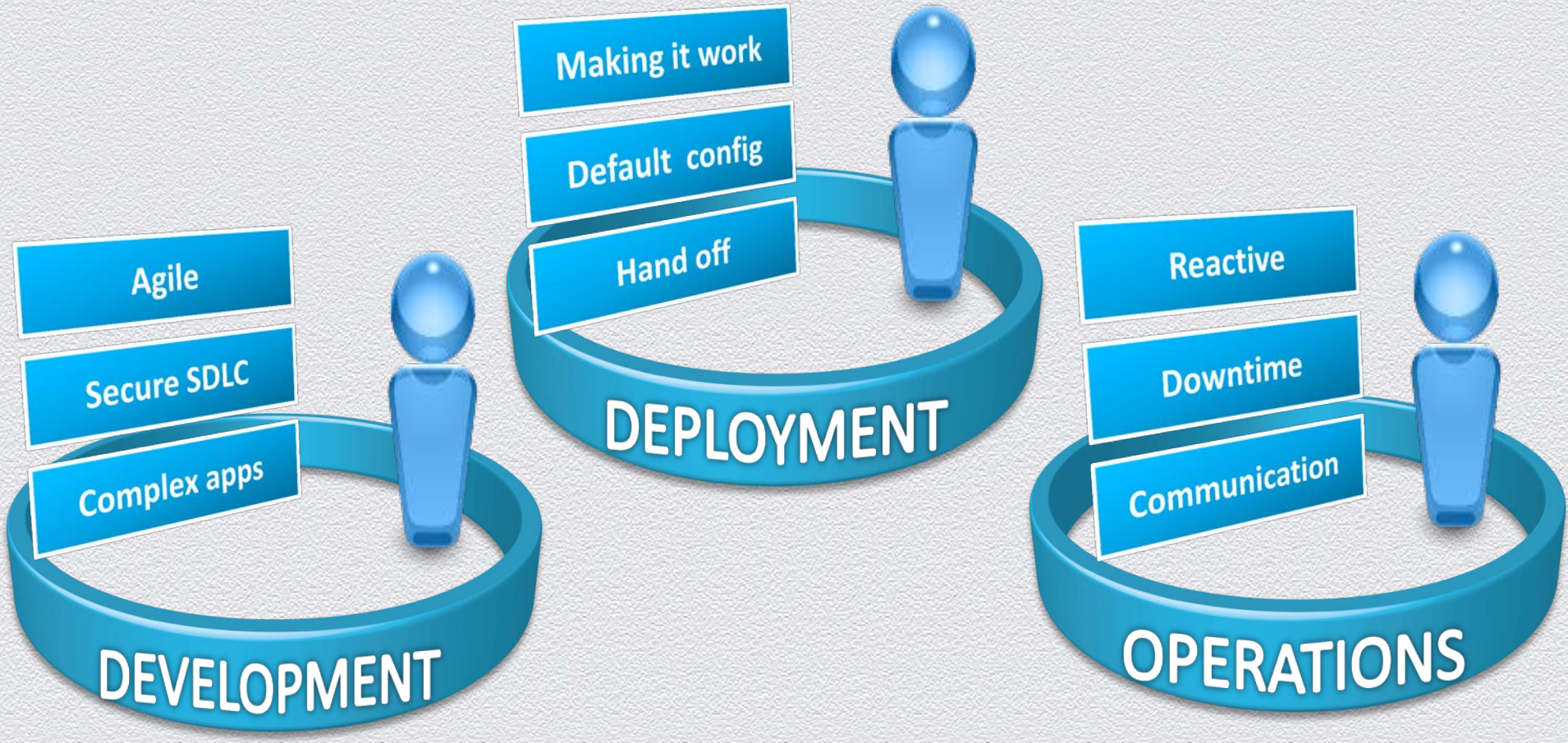
Shaik Mokhinuddeen

Director, Software Engineering
CA Technologies

Ravindra Rajaram

Principal Software Engineer
CA Technologies

Development – Deployment – Operations

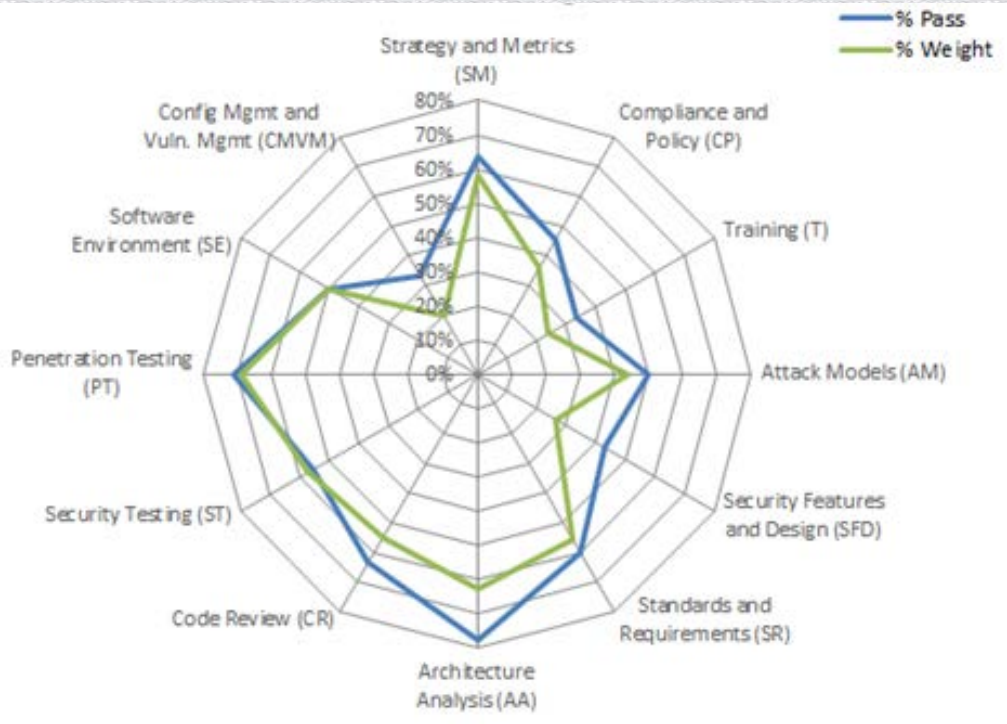


Development - Challenges

- Non-compliant Secure SDLC
- Adopting Agile
  - Many “working” releases
  - Many assessments!
  - Harder to keep track of “issues”
- Bigger the complexity, longer it takes for a thorough assessment
- Development practices
  - Finish the sprint; lets worry about secure coding practices in the “last” sprint
- Lazy APIs

Development - Recommendations

- Strong Secure SDLC
  - BSIMM
    - Open standard
    - BSIMM score card
- Custom implementation
  - Policy & Procedure
  - Assessments
    - Code analysis
    - Penetration testing
    - Design analysis
  - Vulnerability management
    - Internal
    - External
  - Training and Awareness



Development Recommendations

Contd..

- Well defined checkpoints and processes
- Hold product owners accountable for security checkpoints
- Not going to happen overnight
- Implement levels of acceptability
  - Split the assessments into Standard and Advanced
  - Warn the engineering on judicious use of tools
- Training programs at all levels
- Make the product owner accountable to run “part of the” assessment

Development Recommendations

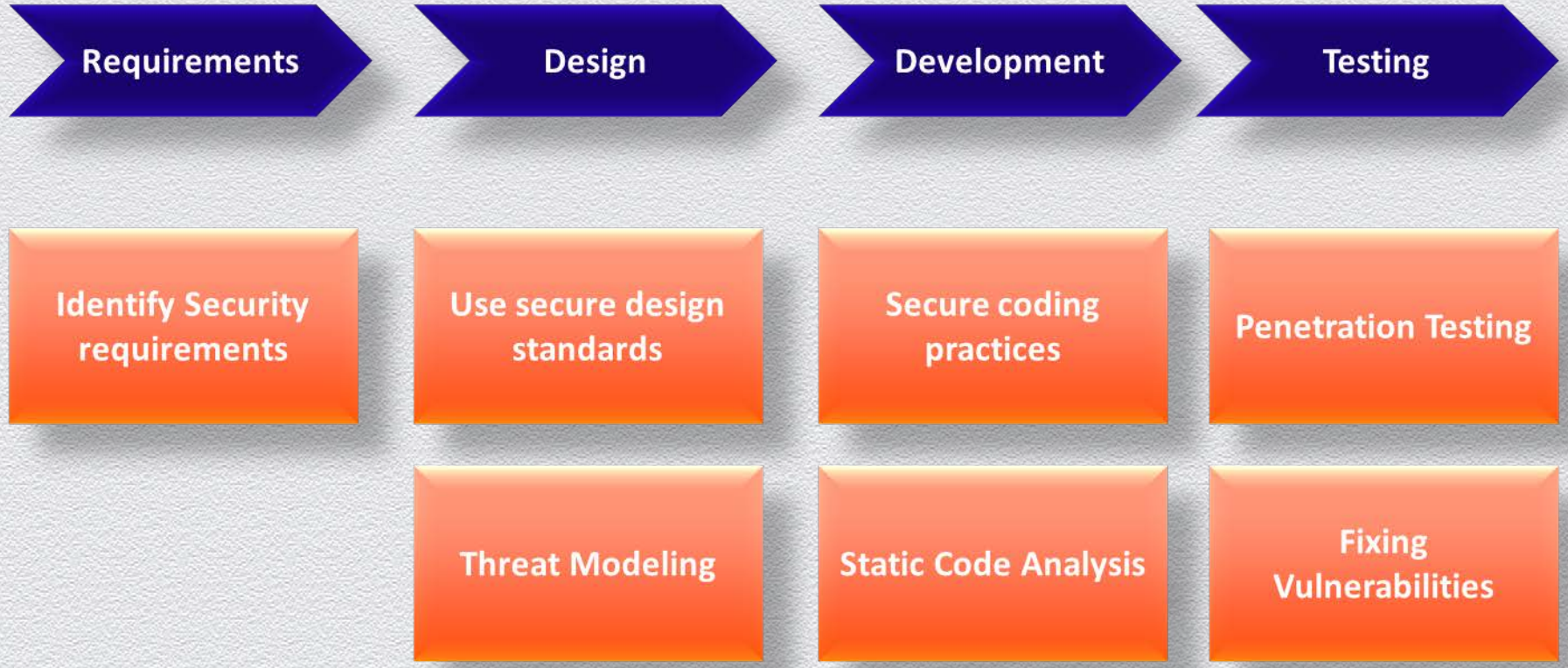
Contd..

- Architects
  - Recommended blueprint
  - Compliance requirements
  - Audit requirements
  - Security requirements
- Track 3rd party software use
  - Useful when a vulnerability is disclosed

Development Recommendations

Contd..

Checkpoints for Secure SDLC



Deployment - Challenges

- "Making it work" becomes the top priority
  - Always an "integration" : In-house/licensed
  - Never works out-of-the-box
  - Custom code: I didn't write that code!
  - Is it a web service? Never mind, open that port!
- Security takes a back seat
  - Security is "out of the window" between Load balancer and App Server
  - Default passwords/configuration
- Security assessment "just before" deployment
  - Delays in getting to the market

Deployment - Recommendations

- Include the “Deployment” team in defining the blueprint of the deployment architecture
- Identify and arrive at the best possible deployment scenario for Penetration testing during Secure SDLC
- Document secure configuration best practices for application servers, database servers etc,.
- Hand-over the reports from Secure SDLC assessments to be shared as necessary for compliance and auditing checks
  - Central location with limited access for all the reports
  - Custom report templates to share internally and externally

Deployment – Recommendations

Contd..

- Tools for security assessment
- Common platform for assigning severity of the issues
- Priority channel for communication between
  - Engineering team
  - Secure SDLC team
  - Deployment team
- Custom code should be shared to the engineering
  - To be included as supported features

Operations - Challenges

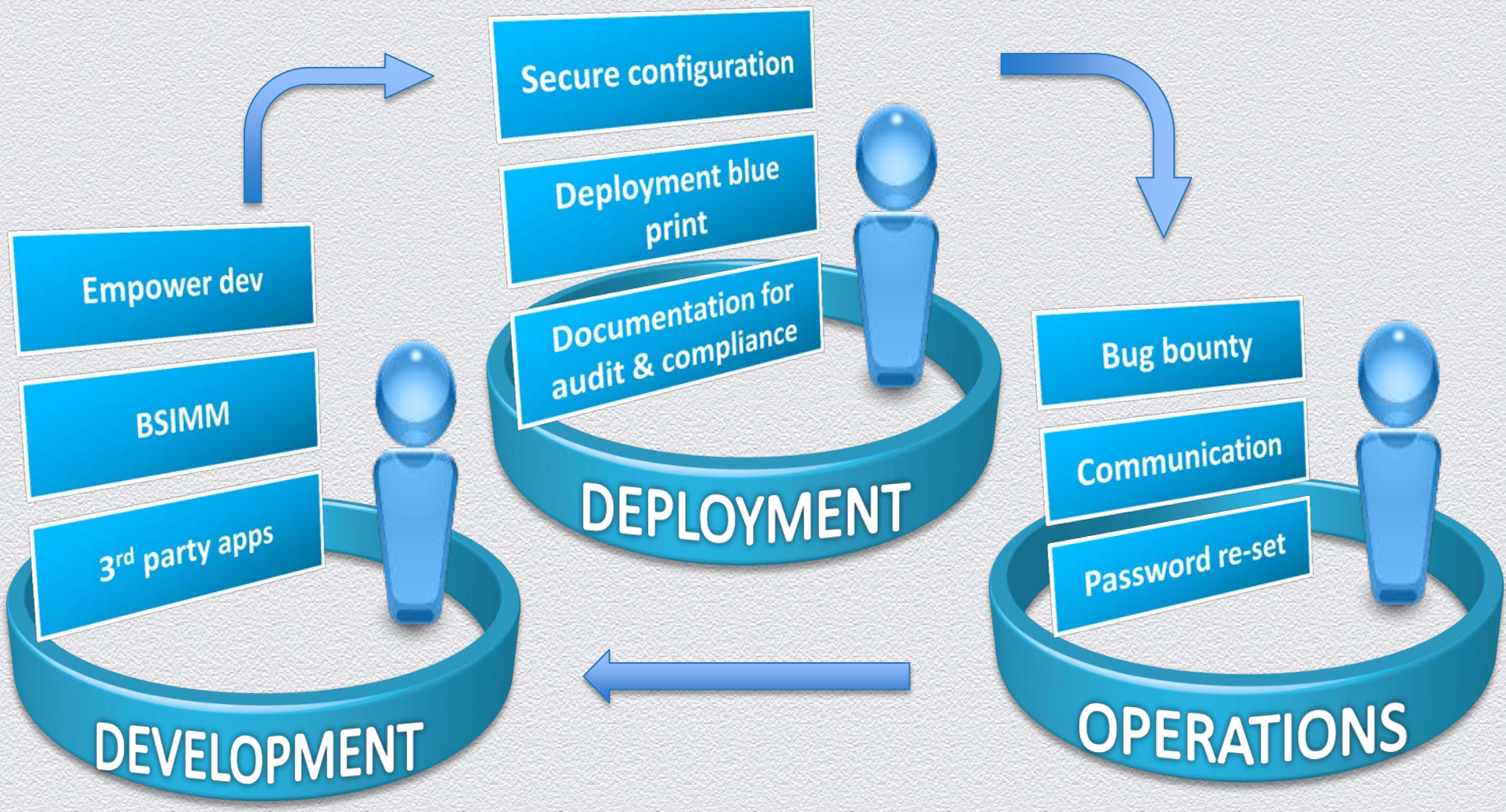
- Highly reactive
- WAFs not very effective
  - Data formats
  - Signatures
- Keeping it quite!
  - Internal leaks
- Downtime
- Not enough Bug Bounties
- Password re-set problems

Operations - Recommendations

- Bug bounties
- Response team
  - Action plan
  - Advisory - immediately
- Well defined “mass” password re-set policy
  - Less response time
  - Block logins for affected users
  - E-mail communication with reset links (possibly with a two-factor authentication)
  - Idiot-proof technology
  - Fool-proof approval process
  - Limited access to the “shiny red button”

Recap

- Thorough process of Secure SDLC
  - Proper checkpoints
  - Empower the developers
    - Training
    - Tools
- Handoff of Secure SDLC artifacts to the Ops team
- Data, Data & Data
  - Use a systems management software that uses analytics to help you be proactive
- Password re-set
  - Predefined policy



Thank you