Microsoft® IT

Microsoft Security Development Lifecycle for IT

Rob Labbé
Application Consulting and Engineering Services
roblab@microsoft.com

The Reasons for Secure Software

There are many threats to data and systems

- Data can be stolen by attackers
- Data can be corrupted by viruses
- Data can be lost or corrupted by employees
- IT Systems can be used by attackers
  - To send Spam, viruses, or launch other attacks from
- IT Systems can be crashed by attackers

Application Layer = Weak Point

- Attackers target the weakest point. The OS Layer and Network layer are too hard now
- On Average over 70% of IT security budget is spent on Infrastructure, yet over 75% of attacks happen at the Application level
- According to Microsoft research, only 1/3 of developers are confident that they write secure code
- The focus must be on hardening the application layer

Reasons for IT Security

- Card Services - A Real World Example
  - $10 Million a month in revenue
  - Processed credit cards for American Express, Master Card, and VISA
- They Lost 40 Million records to hacking
  - Government imposed heavy fines
  - Subject to audits every 6 months for 20 years
  - Amex, Master Card and Visa dropped them
  - A $10 Million a month company destroyed

Hackers are very smart

We need better security

Implement an SDL

- Implement an SDL to build security into your development process
- Train your developers in secure coding techniques
- Incorporate Threat Modeling, Secure Code Review, Security Focused Testing into the process

Purpose of the SDL

- Inventory and assess applications
- Identify and ensure resolution of security/privacy vulnerabilities found in those applications
- Enable Application Risk Management:
  - Strategic
  - Tactical
  - Operational
  - Legal

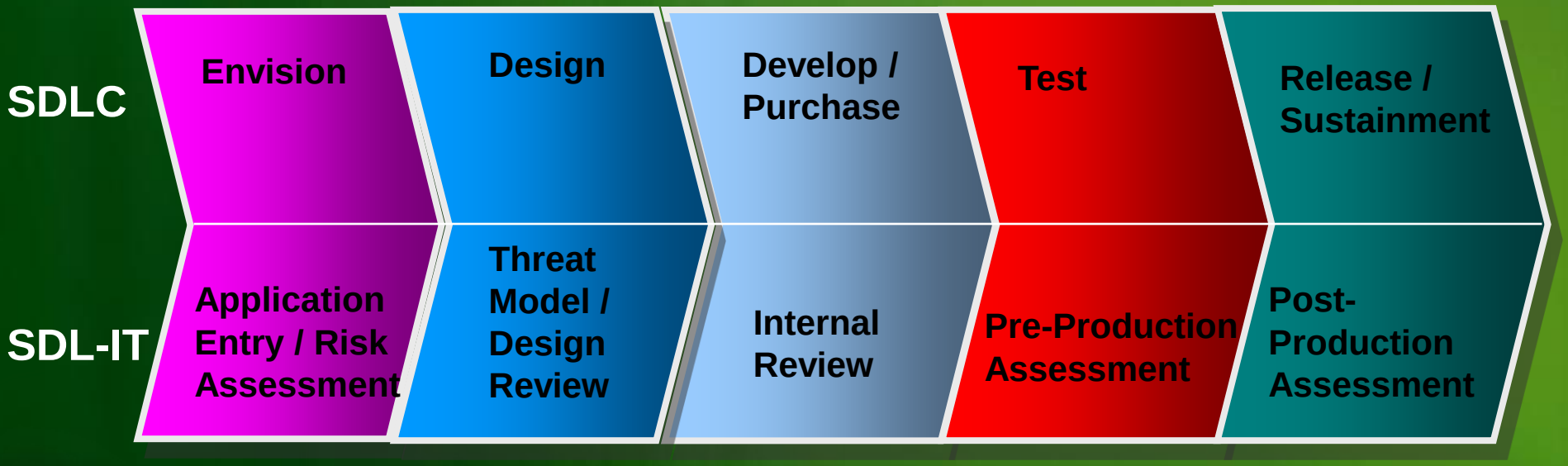
The SDL is NOT Optional

- At Microsoft all line-of-business application teams must go through SDL-IT, All shrink-wrapped products must go through the SDL
- If they fail to do so, they cannot go into production
- Enforcement of the SDL-IT process attributes to it’s success

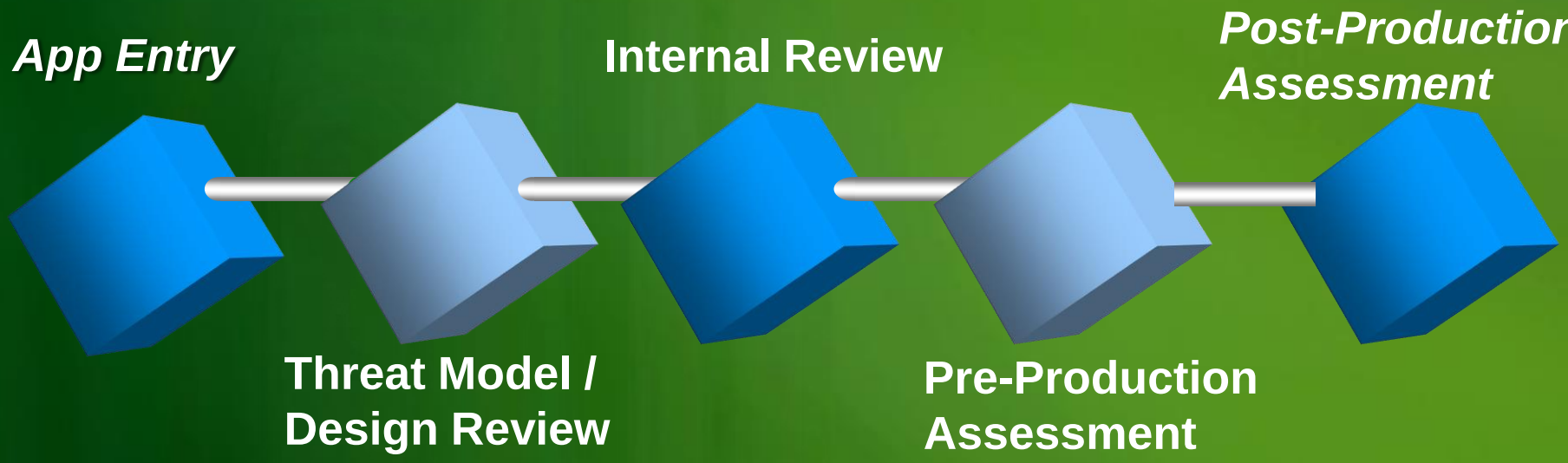
Visibly Measure the Process

- Have internally visible score cards
- Have contests to see if you can find bugs and offer prizes
- Offer incentives to teams with the best security records (no cheating!)

SDL aligns with SDLC

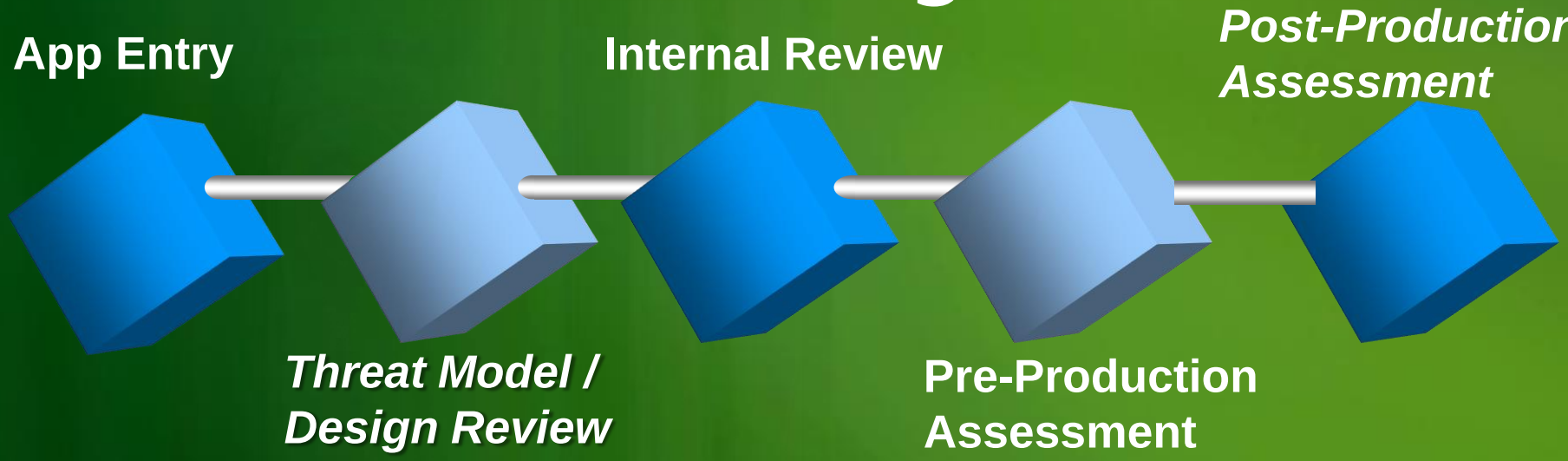


Application Entry / Risk Assessment



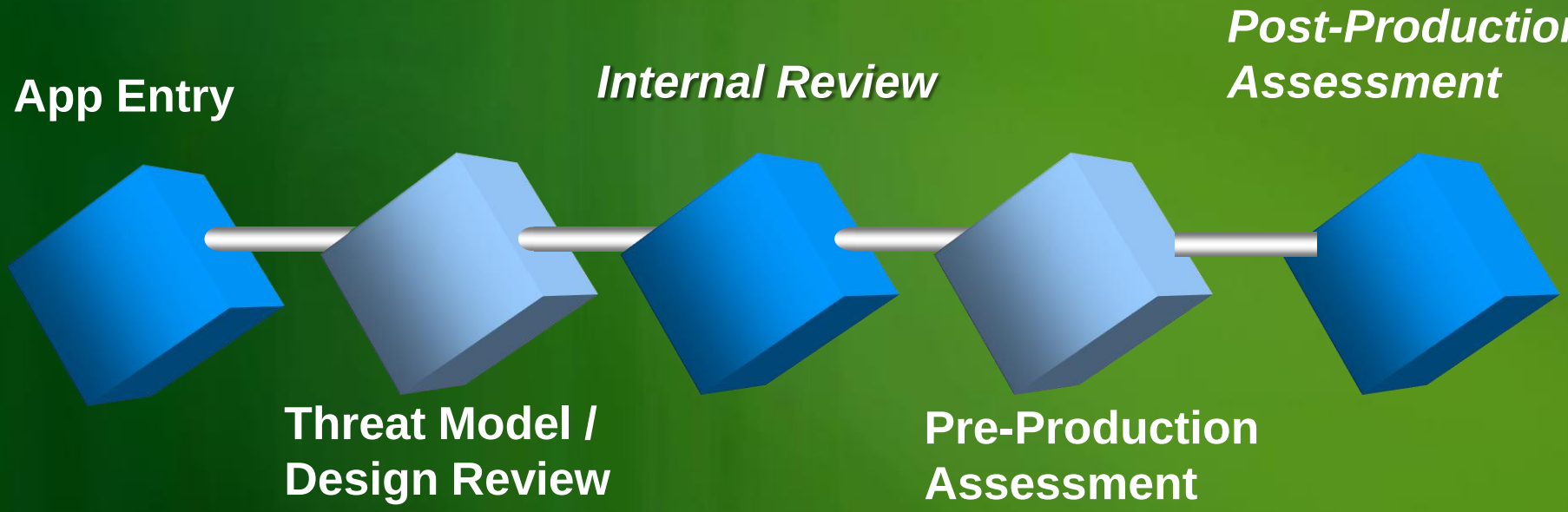
- Objective:
  - Application Inventory
  - Determine Application Risk Categorization
    - High Risk Security/Privacy Release
    - Medium Risk Security/Privacy Release
    - Low Risk Security/Privacy Release

Threat Model / Design Review



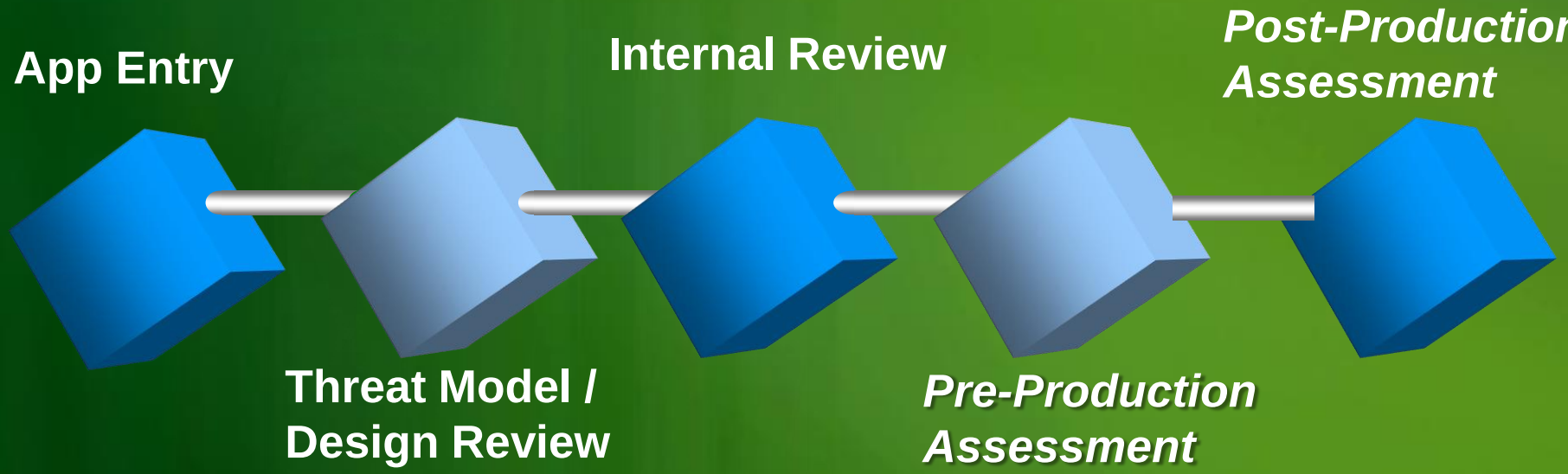
- Objective:
  - Threat modeling provides a consistent methodology for objectively evaluating threats to applications.
  - Review application design to verify compliance with security standards and best practices
  - Verify application meets application principles
    - Confidentiality
    - Integrity
    - Authentication
    - Authorization
    - Availability
    - Non-repudiation

Internal Review



- Review security checklist/policy site
- Team conducts ‘self’ code review and attack and penetration testing

Pre-Production Assessment



- Objective:
  - Low Risk Applications
    - Host Level Scan
      - Windows
      - IIS
      - SQL
  - High/Medium Risk Applications
    - Host Level Scan
    - White Box Code Review

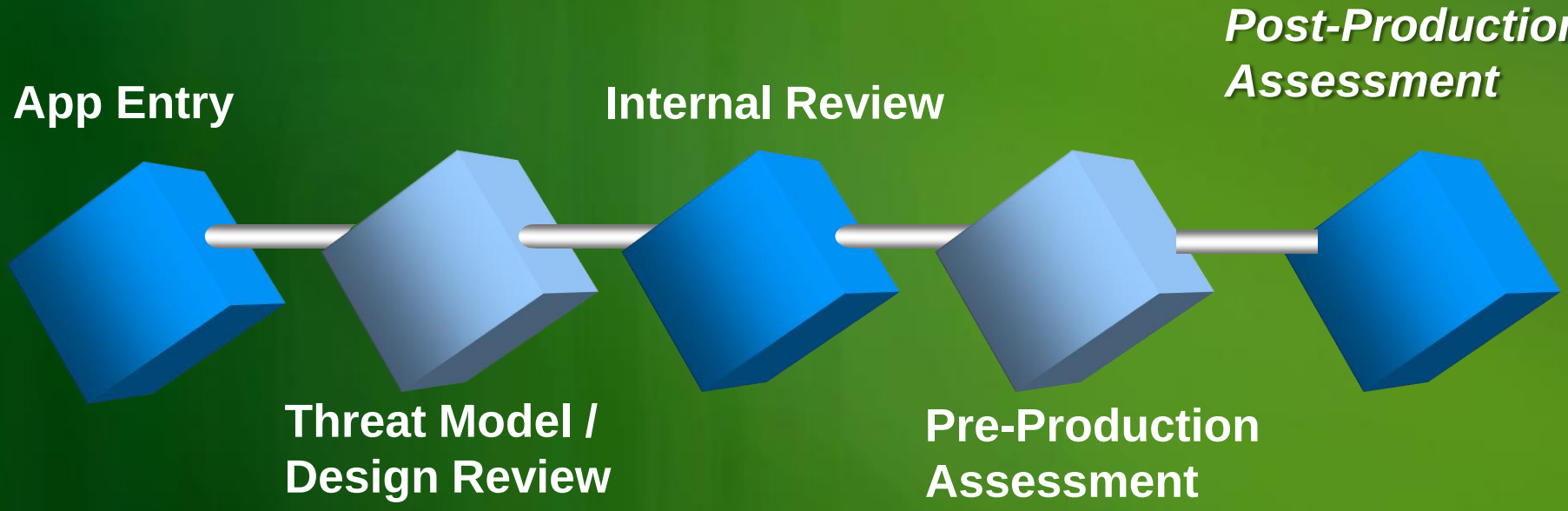
White Box Code Review

- Process
  - Application team provides source code
  - Analysts review application code uncovering security vulnerabilities
  - Vulnerabilities logged in bug database
  - Application team required to address all sev 1 bugs prior to going into production

Some common attack patterns white box review may reveal

- Cross-Site Script Vulnerabilities
- SQL Injection
- Buffer Overflow
- Poor Authorization Controls
- Secrets Stored In Clear Text

Post-Production Assessment



- Objective:
  - High/Medium/Low Risk Applications
    - Host Level Scan
      - Windows
      - IIS
      - SQL

Conclusion

- The need for security is obvious, we have to protect the company and our customers
- To do that we need
  - Management Support
  - Secure Development Life Cycles
  - Developers trained in secure development
  - A Security First attitude!

Conclusions

- Continuous improvement of the process
- Invest time in upfront activities:
  - Threat Modeling
  - Design Reviews
- A holistic view
  - People
  - Process
  - Tools
- It may seem hard to get started – ask for help!

People: Providing guidance on secure application development

Process: Security cannot be an afterthought

Tools: Providing the most innovative tools

Call To Action

- Implement a Secure Development Lifecycle
- Create more secure and reliable software
- Build Trust!

Resources

- ACE Team Blog:
- http://blogs.msdn.com/ace_team/default.aspx
- Threat Modeling Tool
- http://go.microsoft.com/fwlink?linkid=77002
- Threat Modeling Blog:
- http://blogs.msdn.com/threatmodeling/
- Rob Labbé's Blog:
- http://blogs.msdn.com/roblabbe/default.aspx

Microsoft® IT

Microsoft®

Your potential. Our passion.™