The Insecure Software Development Lifecycle

How to find, fix, and manage deficiencies within an existing methodology

#InsecureSDLC

April C. Wright
The Status Quo of Software Development Lifecycles

• Concern with Speed and Cost, not Quality
• Reliance on offensive testing to fix security issues
• Security often “tacked on” to software
• Risks arising from software creation/integration are not fully understood
• Builders (Developers, Integrators, DevOps, etc) are not knowledgeable about security methods of attack and defense
Stakeholders

“A stakeholder is any person or group that affects or is affected by a particular project. Along the path to completing your project, stakeholders can be partners, resources, or roadblocks—and potentially all three rolled into one. Stakeholder buy-in, the cooperation or positive participation of a stakeholder, is the preferred condition for any successful project.”

-- Scott Shpak

http://bit.ly/2K59RB0
Understanding Stakeholders and Existing Processes

• We must understand these to elicit successful change
• Process mechanics != People
• Computers don’t resist change
• Historical data
• Observation
• Inquiry
Relevant Stakeholders

- Designers
- Architects
- Builders / Developers
- Implementers
- Operators
- Organizers
- Executives/Board
- Customers
- The organization itself
- Third parties / Vendors
- Auditors

http://bit.ly/2K59RB0
### Stakeholders have differing points of view

<table>
<thead>
<tr>
<th>Security's goals:</th>
<th>Builder's goals:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create it securely</td>
<td>Time to market</td>
</tr>
<tr>
<td>Maintain it properly</td>
<td>Profit</td>
</tr>
<tr>
<td>Prove it’s protected</td>
<td>Correctness</td>
</tr>
<tr>
<td>Documentation</td>
<td>Minimal defects</td>
</tr>
<tr>
<td></td>
<td>Optimization *</td>
</tr>
</tbody>
</table>

* (Chuck Norris writes code that optimizes itself)
Project Managers are great assets

- Supervising adults
- Master plan
- Manage schedule, get things done
- Caveat: Garbage in/Garbage out
- Help build security in
QA and DevOps

• Can make use of automation for testing security
• Ensure requirements are met
• …Assuming you’ve provided requirements!?
Legal

- Privacy
- Compliance
- Confidentiality
- Liability
- Risk
- Interests are very similar to Security’s!

HOLD MY CALLS

I AM WORKING ON A TOUGH CASE
Customers/End-Users

- Privacy
- Compliance
- Contract
- Cost
- Trust
  - Repeated violations = loss of trust
  - Fairly easy to start with trust
  - Difficult to regain
3rd Parties

• Supply chain is critical
• Each part of supply chain has links to other parts of chain
• Chain is only as strong as its weakest link
• Policies, compliances, needs vary
• Security varies
• Risk tolerance varies
• Auditable
Analyzing existing processes

- Interviews – ask questions, seek explanations
- Observations – stakeholders, key players, motivators, drivers, blockers
- Process and procedure analysis – review complete process start-to-finish
- Metrics – what is being tracked? What *should* you track?
- Find gaps – what is not being done that should, what could be improved
- Document gaps – for reporting: rating and prioritization
Gap Analysis

• Do you have a risk management program?
• Are there single points of failure?
• What is bad?
• What is actually good?
• Avoidable delays?
• Security milestones not being set/met?
Document the gap analysis

- Summarize existing process for those new to it
- Describe state of security
- Document good aspects
- List gaps, explain risk
- Show gaps chronologically within the process timeline (for now)
How does security affect the stakeholder?

• Cognitive bias – status quo is ok! even if it is only ok!
• Change is uncomfortable
• Uncertainty manifests as **physical pain** in the brain
• Everyone has goals – role-based, individual, team-based
• Appeal to their goals
• Acknowledge and empathize
How does security affect the process?

• Document any undocumented processes
• Where should Security provide input?
• Where should Security be a checkpoint?
• Collaborate with other stakeholders
• Maturity model
Preparing for rebuilding the program

YOU NEED A PLAN!
The secure end-state must feel necessary to the org
How are you going to achieve the goal?
Key program metrics

- Number of security bugs trending over time
- Types of security bugs found
- Comparisons of number of security bugs versus other bugs
- Severity of security bugs trending over time
- Regressions seen between releases
- Recurrence of similar types of security bugs (e.g., buffer overflow conditions found repeatedly)

SANS “Using Metrics to Manage Your Application Security Program”
Metrics

- Metrics help identify gaps
- Metrics help with attribution of gaps
- Are existing metrics adequate?
- Must be acted upon, not just reported!
- Track best and worst (reward best, train worst)
- Want to show time NOT spent, decreases in cost
- Will help explain ROI to management

SANS “Using Metrics to Manage Your Application Security Program”
Important metrics

For software security to be a priority, CxO’s need to understand (from SANS):

- Improvements overall
- Improvement to availability / operational risk
- Reduction of delays to delivery
- Reduction of cost of operations
- Implemented risk reduction techniques
- Residual and unmitigated risks
- Threats to customers/company

https://www.sans.org/reading-room/whitepapers/analyst/metrics-manage-application-security-program-36822
Phased goals

• “Rome was not built in a day”
• Create 3 phases for gap closing and other goals:
  • Phase 1 – ‘low-hanging fruit’/easy-win and any critical gaps that must be changed before anything else
  • Phase 2 - gaps that are important but cannot or should not be addressed until you address the Phase 1 gaps
  • Phase 3 – Strategic long-term change that requires planning, resources
Goal phases

- Phase 1 – Introduction to change
- Phase 2 – Using existing resources more effectively, active support and participation
- Phase 3 – Longer term goals, not necessarily “last” phase, ‘where the program is going’
- Phased plan will be presented to Management
Gaining management support

Management helps set expectations with other stakeholders and provides support when there is reluctance to cooperate.

1. Gap assessment
2. Phased goals
3. Prioritized and ranked gaps/goals

All = Long Term Plan
Gaining management support

- Objectively gathered Metrics = influential
- We need to communicate risks:
  - Harm to the brand: Invaluable
  - Legal implications
  - Bug found – cost to fix:
    - Requirements phase $1
    - Design phase $5
    - Dev phase $50
    - Testing phase $500
    - Found by attacker: $ Millions
- Risk Management results and scoring
Planning requirements

- What needs to be done
- Who needs to be involved
- Costs or resource shifts necessary
- Why the plan is important
Active stakeholder participation

• Participative decision making (PDM) fosters an environment in which people have a choice whether to be motivated and contribute
• Actively include stakeholders in all decisions related to the Plan
• Engagement = inevitable support
• Stakeholders need to understand why they are involved, what is expected of them, and how their contributions are valuable

Working as a unified team (but not much for the business)
Working as a unified team

Purple Team / Red Team without defensive building:

• Does not address upstream source of vulnerabilities?
  • Where are security bugs coming from?
  • Why are there so many?
  • Root-cause analysis might not be performed.
• Inherent delay between inception and identification of a vulnerability
• Looks for known vulnerabilities and exploiting weaknesses in unpatched systems
The importance of collaborating as one team

- Partnering, not policing
- We all want to be successful & want the org to be successful
- Communication solves a lot of problems
- Challenge of one is actually knowledge of another
- Builders are on the front line of defense
- Reaching out with questions yields better software
Discussions, not just bug submissions

- Detailed meetings to discuss findings from offensive testing
- Review nature of vulnerabilities
- Ask questions that encourage critical thought
  - Did you use this code anywhere else? Out of scope?
  - Are other people using code like this?
  - Are there similar bugs?
- Deeper conversations, not just tickets for fix
Positive interactions

• Focus on the issue, not the person who caused it
• Be sincere when giving praise
• Acknowledge the effort, not the ability
• Start with a positive comment, then provide constructive criticism, then close with another positive comment
Rotating work assignments and embedded liaisons

- Jumpstarts conversations
- Goal is not cross-training
- Security viewed as “same team”
- Less formal interactions more frequently
- Expect decreased productivity initially, more secure / higher quality over time
Setting expectations for stakeholders

- Expectations need to be set UP FRONT
- Project can plan for increased time spent on security
- Inadequate expectations can result in slipped timelines
- Stakeholders need to be aware of what steps need to be taken if you want those steps to be taken
- PM can help drive the steps to completion
Using organizational policy to create a need

• Mandatory requirements drive change
• Formally recognizes Security as a stakeholder
• Engage stakeholders to agree on terms
• Explain role-based stakeholders / responsibilities
• Language is important
  • “Must” & “Shall” vs. “Should” & “Generally”
• Process for handling exceptions
Using compliance to create a need

- Validation that an org does what it says it does
- Dictates requirement, not “how”
- Audits drive compliance, compliance drives action
- Policy must be followed, so include security in policy!
- Identify end-users of software, tailor requirements in SW
- PM can plan for audits and costs
- May not actually be achievable until later releases
Knowledgeable humans

• Will new processes require training?
• Can the security team take time to offer in-house training?
• Is the security team equipped to train others?
• Is there training that you can purchase and make available throughout the enterprise?
• Should a third-party develop custom training?
The development style guide and standard libraries:

- Policy for developers for consistency in coding
- Detail common interactions between software functions, and define:
  - How and when to use comments
  - Tabs or spaces for indentation (and how many spaces)
  - Appropriate use of white space
  - Proper naming of variables and functions
  - Patterns to be used / patterns to be avoided
Style guides

- Sometimes known as a ‘pattern library’
- Originally used in print design for achieving consistency of logos and imaging
- Guides have become useful for coding
- Method for ensuring development consistency/quality across projects/teams
- Standardize design and build process
- Consistent code
Automated code scanning vs Manual code reviews

• Serve different purposes
• Automation:
  • Like “spell check” – similar limitations
  • Finds common bugs easily
  • Good first phase goal
  • False positives/negatives expected
• Manual reviews find logic errors, flaws (compare to bugs)
1100
1525
1545

Started

Cosine Tape (Sine Test)

Started Mult + Adder Test

Relay #7

(moth) in

First actual case of bug

1630

1700

Arctangent started.

Closed down.
Checklists set and monitor expectations

- Start with desired end-state, work backwards
- What does it look like, how does it behave?
- Who has vetted its functionality?
- What does it do and, more important, what does it not do?
- New software vs Existing software updates/upgrades
- Provide as early as possible to PM
- Store centrally for life of software
Conclusion

• We usually do not have the luxury of starting from scratch
• Identify and manage stakeholders (blockers or advocates)
• Catalog gaps – rate, prioritize, plan
• Injecting security early cuts down cost to fix things later
• Small wins lead to bigger changes
• Phase-in changes to make them easier to accept
• Stakeholders in a decision are more likely to support it
• Empathy and understanding go a long way
The Insecure Software Development Lifecycle

by April Wright for Hack In Paris

@aprilwright

#InsecureSDLC

Multumesc!