RSAConference2016
San Francisco | February 29 – March 4 | Moscone Center

Connect to Protect

SESSION ID: ASD-R04

Agile Security – Field of Dreams

Laksh Raghavan
Senior Security Strategist
PayPal Inc.

@laraghavan

The views expressed in this presentation are my own, and not those of PayPal Holdings, Inc. or any of its affiliates.

#RSAC

Introduction

- This presentation is:
  - A case study on how PayPal's Secure Product Lifecycle (SLPC) had to adapt to Agile
  - Vendor neutral
  - Descriptive
- This presentation is NOT:
  - Silver Bullet ™
  - Sales pitch
  - Prescriptive- if you implement, your end product may vary

PayPal's Agile Transformation

- Some interesting stats and facts about our Agile Transformation:
  - Big Bang approach against prevailing wisdom
  - Went from project driven to product aligned
  - 400+ scrum teams across the globe
  - 500+ Change Champions and 165 Transformation team members
- Every "industry expert" we consulted told us we couldn't transform at this scale in our designated timeline but we did it!

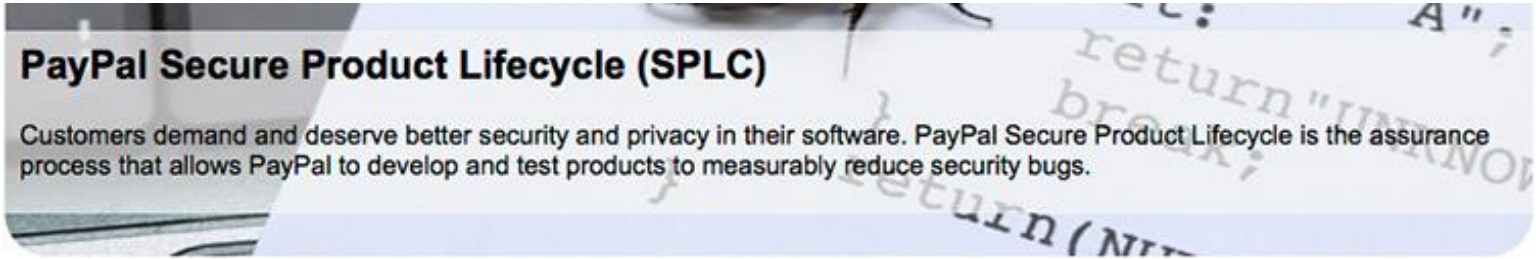
Setting the Stage for Change

- The PLC process was complicated and inefficient:
  - A nearly 50 step process
  - Separate standards /procedures as PDFs/HTML on an internal portal
  - Manual gates in the lifecycle and human involvement for all projects
  - Threat Model EVERYTHING

It was great for security because it, (a) bought us time and (b) created a lot of documentation for review and future reference. But alas…

When the president of the company says you'll become more agile, you do it (no matter how scary).

PayPal SPLC - Overview



- Objective:

Reduce the number of vulnerabilities in our products over time by building repeatable/sustainable proactive security practices embedded within our PLC.

SPLC Transformation

- **Strategy**
  - Institutionalize risk-based thinking and processes
  - Secure by Default – Frameworks, Dev. Tools, etc.
  - Put our bots to work
- **Execution**
  - People – Internal PD security champions to help drive focus and attention on software security
  - Process – Integrate seamlessly with our “agile” way of delivering products.
  - Technology – Secure frameworks, libraries and automated tools that enable PD to ship products rapidly ***and*** securely

Typical Challenges

- Multiple product teams
  - Various time zones
  - Different priorities/risks
  - Multiple coding languages/frameworks
- Agile vs. Waterfall
  - User Stories vs. Comprehensive Documentation
  - Minutes/Days vs. Weeks/Months

My journey and ”Aha” Moment

- My initial perspective of Agile was:

- Field of Dreams: “If you build it, they will come”
- But soon we realized its benefits, too.
  - A reported XSS issue was fixed in minutes rather than hours/days

How did we adapt to the change?

An exercise in testing (and trusting) the automated process

- Dynamic/In-Context Security Requirements: Security Stories
- Automated controls in the lifecycle
- Secure Frameworks and Security Tools used for all projects & human involvement for critical-risk projects
- Threat Model only things that aren't run-of-the-mill web or mobile apps and/or not built on our standardized secure frameworks

What were our guiding principles?

Security Options Auto-enabled to Protect Developers by Default

- If we rely on *__every__* PayPal developer doing the right thing from a security perspective *__every__* time he/she writes code, we are doomed to fail!
- Wherever possible, security controls are to be made available automatically and turned ON by default
- Developers have go out of their way to turn off security controls
- Secure-by-default in all layers
  - Perimeter
  - Infrastructure
  - Framework
  - Libraries
  - Dev. Tools
  - Code/Config

Stronger reliance on automation

- In the world of CI and CD, automating security is your key to success
- Developers should NOT have to learn a new tool or go out of their routine dev. tasks and tools to invoke security
- Security tools must be seamlessly integrated into our development tools
- Automated monitoring and reporting for deviations from our security standards
- No need to manually file tickets for issues found by developers within the lifecycle – *less* red tape and *more* work done!
- That doesn't mean that there'll be no measurement: Invest in automated metrics gathering and compliance tracking

Federated model that enables scaling

Our landscape is huge!

- A central InfoSec team with just handful of AppSec experts can never scale by themselves
- We needed a federated model of security champions across all delivery groups who can be our eyes and ears:
  - Responsible to ensure their products follow SPLC practices
  - First point-of-contact for SPLC within the product delivery groups
  - Trained and Knowledgeable on our SPLC processes and tools
  - Know who to loop in from InfoSec for what and when
  - Help triage and fix security vulnerabilities

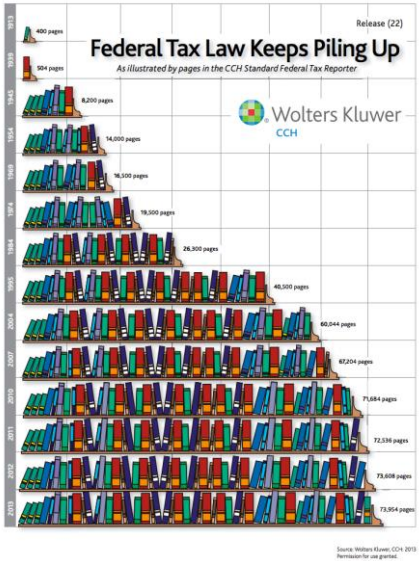
So, what did we really change?

Key Change #1: Security Stories

Holy Grail for any software security professional ➔ Make functional and non-functional requirements equal partners

In Agile Speak: Make User Stories and Security Stories equal partners

Before:



After:

Your Favorite Tax Software!

Key Change #1: Security Stories, cont.

- A web-based tool that hooks into the Release Planning (aka Multi-Sprint Planning) process
- A simple survey that does light-weight threat modelling, generates security stories, and places them in the backlog of the scrum team
- Tracking and reporting from within our Agile LifeCycle Management tool

Key Change #2: Automated controls in CI/CD

- Hook security checks into the CI/CD processes
- Start with a warn-only mode and in parallel do your communication/awareness and training sessions to help your developers comply to your security requirements
- Shift to block mode in a phased manner, as needed
- When their build turns red or breaks, developers pay attention and fix what they have to!

Key Change #3: Security Champions

- Security Champions across all product teams
- Started as a volunteer program
- 80-20 rule in action
- Transformed the program in 2015 to bring in “dedicated” champions with goals baked into annual performance reviews
- Train-the-trainers
- Drive the culture change!

And what were the results?

Show me the money!

- Measurable improvements in compliance to security requirements
- Our modern frameworks and apps built on those frameworks are inherently way more secure than our legacy stacks
- Improved SLA adherence for fixing application security bugs
- Early engagement means no or minimal projects hit security roadblocks during launch
- A quote from our Android App's Team Manager:

  "It is great to know that the pentest didn't find any blockers and it can be largely attributed to the fact that we are following SPLC…"

In a Nutshell

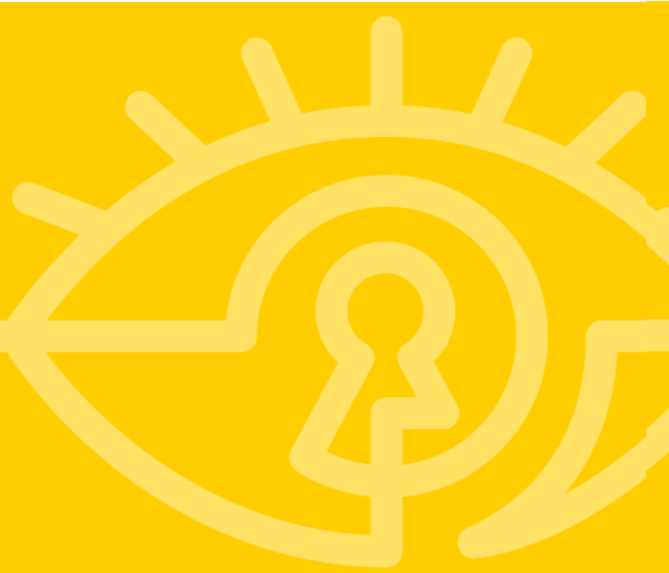
Legacy SPLC	Agile Transformed SPLC
200+ PDF/HTML security standards and procedures	Security Stories customized for the scenario
Manual gates throughout lifecycle	Lifecycle relies on automated controls
Human involvement for all projects	Let the frameworks and tools do the heavy lifting - human involvement for critical risk projects only
Threat Model everything	Threat Model only critical scenarios

Challenge everything to drive change!

Apply What You Have Learned Today

- By next month you should:
  - Understand how Agile works at your organization and its maturity level
  - Critically analyze your existing SPLC practices
- In the six months following this presentation you should:
  - Build a roadmap for SPLC transformation
  - Start small and experiment the changes with a few small teams
- Within a year you should:
  - Pilot with large business unit(s)
  - Take the lessons learned from the pilot and tweak the process/tools for org-wide rollout

Questions?



Thank you!