.conf2015

# Developing at the Speed of Splunk

Kurt Chase

Director of Release Engineering, Splunk

Itay Neeman

Director of Engineering, Splunk

splunk>

# Disclaimer

During the course of this presentation, we may make forward looking statements regarding future events or the expected performance of the company. We caution you that such statements reflect our current expectations and estimates based on factors currently known to us and that actual events or results could differ materially. For important factors that may cause actual results to differ from those contained in our forward-looking statements, please review our filings with the SEC. The forward-looking statements made in the this presentation are being made as of the time and date of its live presentation. If reviewed after its live presentation, this presentation may not contain current or accurate information. We do not assume any obligation to update any forward looking statements we may make.

In addition, any information about our roadmap outlines our general product direction and is subject to change at any time without notice. It is for informational purposes only and shall not, be incorporated into any contract or other commitment. Splunk undertakes no obligation either to develop the features or functionality described or to include any such feature or functionality in a future release.

# Introductions

## Itay Neeman

Director of Engineering, Data Platform

- Focus on backend features, e.g.:
  - KV Store
  - HTTP Event Collector
  - Distributed Management Console
- Focus on Developer Enablement, e.g.:
  - SDKs
  - Tools
- At Splunk for 4+ years

## Kurt Chase

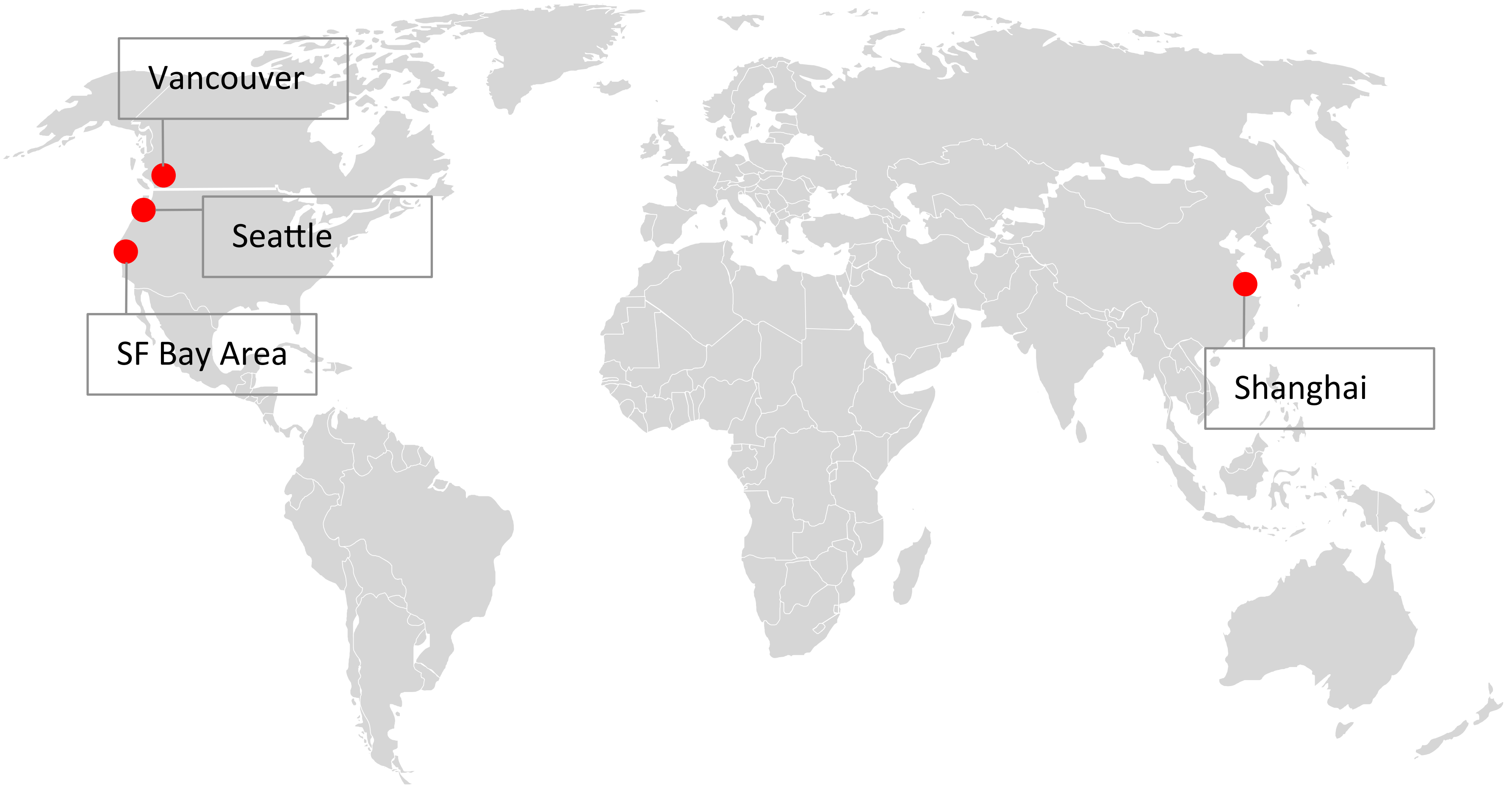
Director of Release Engineering

- Global Release Engineering and Management:
  - Infrastructure
  - Builds
  - Tools
  - And much more!
- At Splunk for 1 year

# Agenda

- Overview of Product Development @ Splunk
- A “day” in the life of a feature
- How we use Splunk to help build Splunk
- Discussion/Q&A

# Products @ Splunk


Vancouver
Seattle
SF Bay Area
Shanghai

# Product Functions

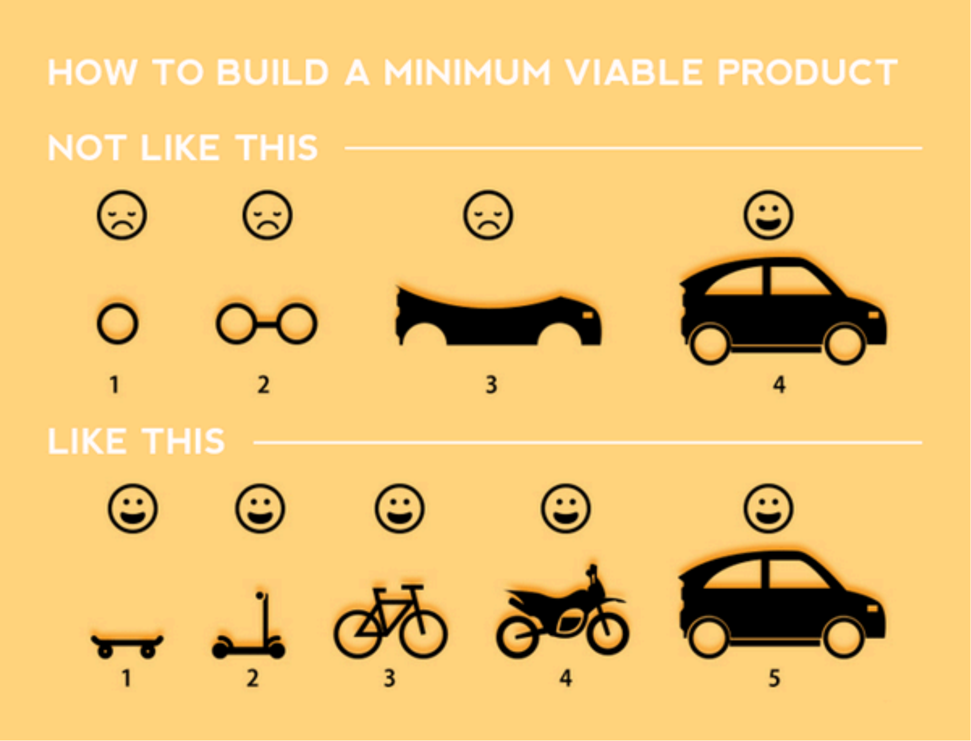
- Product Management (PM) – “What should we build?”
- Quality Assurance (QA) – “Does it work?”
- Development (Dev) – “How do we build it?”
- User Experience (UX) – “How to make it usable?”
- Program Management (PMO) – “Will it be built on time?”
- Release Management (RE) – “What infra do we need to build it?”
- Documentation (Docs) – “How do we communicate how to use it?”
- Product Security (ProdSec) – “Is it secure?”

# Multiple Products

- Splunk builds multiple products:
  - Splunk Enterprise
  - Hunk
  - Splunk Light
  - Enterprise Security
  - IT Service Intelligence
  - Splunk Mobile Access
  - Cloud
- Products drive and influence each other
  - ES and ITSI drove KV Store features in Enterprise
  - Cloud drove installation ease of ES and ITSI
- Common tooling and infrastructure across locations, products and teams

# Agile

- Fully functional scrum teams
- Sprints go for two weeks
- Continuously iterate on MVP



# Best of Breed Tools

JIRA

coverity

HipChat

BULLSEYE
TESTING TECHNOLOGY

loader.io BETA

Jenkins

ANSIBLE

Confluence

.conf2015

# A “Day in the Life” of a Feature

splunk>

# How is the Salame Made?

# Glenn Has An Idea!

# Ideation and Discussion

- Free-form discussion
  - HipChat
  - Confluence
  - Meetings
- Culminates in a Products Requirement Document (PRD)
  - Get more detailed feedback from all stakeholders

# Planning and Design

- Formulate a plan
  - How many people do we need to make this happen?
  - How long is it going to take?
- Engineers start working on Engineering Requirements Document (ERD)
- QA start working on Test Plan
- Convert plan (stories, tasks, tests) to JIRA

# Execute and Iterate

- Team goes into execution mode:
  - Pick a task
  - Write code
  - Write tests
  - Validate
- Variety of tools:
  - Source Control and Code Review: Git/Stash
  - Continuous Integration: Jenkins and Bamboo
  - Testing: internal tools, PyTest, JUnit, ApacheBench, and many more

# Git and Stash

- Stash is our Git server – main home of our codebase
- Each task is a branch, correlated with JIRA (e.g. feature/SPL-12345)
- Before merging branch:
  - Get feedback through code review
  - Get feedback from Continuous Integration
  - Run any necessary manual validation
- Hooked into all our reporting systems

# Continuous Integration

- Use both Bamboo and Jenkins
- Run on as many platforms as we can (Windows/Solaris/OS X/Linux/etc)
- Various kinds of triggers:
  - Per-commit
  - Per-Pull Request
  - Manual
  - Nightly

# Feedback

- Once iteration nears completion, we want to get feedback
- We work with targeted customers:
  - Setup “advisory board”
  - Demo feature for them
  - Work on getting a test deployment where it makes sense
- Gather feedback and continue iterating
- … or bring back into future ideation

# Releasing

- Merge all the features together (from all the feature branches)
- We then make sure:
  - Everything is working together well
  - Performance is up to our expectations
- Package the release into its final forms (installers, tarballs, Cloud, etc)
- Release it to all of you!

.conf2015

# Using Splunk to Build Splunk

splunk>

# Splunk @ Splunk

- Splunk is used extensively all throughout our process
- We use it to:
  - Report holistically on our release and JIRA
  - Track performance measurements across releases and packages
  - Do performance investigations
  - Track the health of our Continuous Integration results
  - Evaluate Product Security vulnerabilities and status

.conf2015

# Splunk @ Splunk Demos

splunk>

# Summary

- You’ve seen how Splunk does engineering
- You’ve seen how we use Splunk to build a better Splunk for you
- You’ve seen some tools that you can use to help your product processes

.conf2015

# THANK YOU

splunk>