What I Learned About Going Fast at eBay and Google

Randy Shoup
@randyshoup
linkedin.com/in/randyshoup

Background

- CTO at KIXEYE
  - Real-time strategy games for web and mobile
- Director of Engineering for Google App Engine
  - World's largest Platform-as-a-Service
- Chief Engineer at eBay
  - Multiple generations of eBay's real-time search infrastructure

Why Are Organizations Slow?

- Organizational Culture
- Process
- People

Why Are Organizations Slow?

- Organizational Culture
- Process
- People

Quality over Quantity

- Whole user / player experience
  - Think holistically about the full end-to-end experience of the user
  - UX, functionality, performance, bugs, etc.
- Less is more
  - Solve 100% of one problem rather than 50% of two
  - Users prefer one great feature instead of two partially-completed features

Quality Discipline

- Quality and Reliability are "Priority-0 features"
  - Equally important to users as product features and engaging user experience
- Developers responsible for
  - Features
  - Quality
  - Performance
  - Reliability
  - Manageability

Quality Discipline

- Developers write tests and code together
  - Continuous testing of features, performance, load
  - Confidence to make risky changes
  - Catch bugs earlier, fail faster

- "Don't have time to do it right" ?
  - WRONG ☺ – Don't have time to do it twice (!)
  - The more constrained you are on time and resources, the more important it is to do it solidly the first time

Google Engineering Discipline

- Solid Development Practices
  - Code reviews before submission
  - Automated tests for everything
- Single logical source code repository

➔ Internal Open Source Model

  - Not "here is a bug report"
  - Instead "here is the bug; here is the code fix; here is the test that verifies the fix" ☺

Service Teams

- Small, focused teams
  - Single service or set of related services
  - Minimal, well-defined "interface"
  - Vendor – Customer relationships
- Clear "contract" between teams
  - Functionality: agreed-upon scope of responsibility
  - Service levels and performance

Google Services

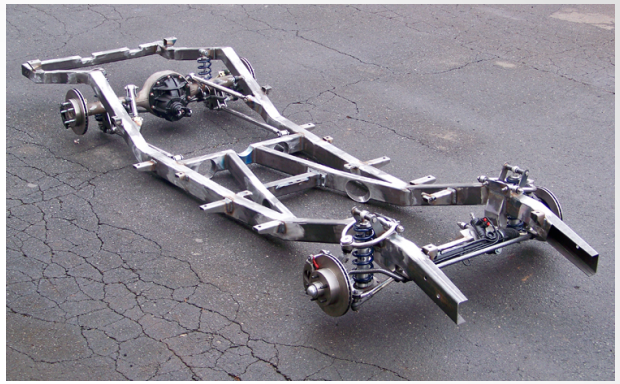
- All engineering groups organized into "services"
  - Gmail, App Engine, Bigtable, etc.
- Self-sufficient and autonomous
- Layered on one another

➔ Very small teams achieve great things

Autonomy and Accountability

- Give teams autonomy
  - Freedom to choose technology, methodology, working environment
  - Responsibility for the results of those choices
- Hold team accountable for *results*
  - Give a team a goal, not a solution
  - Let team own the best way to achieve the goal

KIXEYE Service Chassis



- Goal: Produce a "chassis" for building scalable game services
- Minimal resources, minimal direction
  - 3 people x 1 month
  - Consider building on open source projects

➔ Team exceeded expectations

- Co-developed chassis, transport layer, service template, build pipeline, red-black deployment, etc.
- Heavy use of Netflix open source projects
- 15 minutes from no code to running service in AWS (!)
- Open-sourced at https://github.com/Kixeye

Collaboration

- One team across engineering, product, operations, etc.
- Solve problems instead of pointing fingers

Google Co-Location

- Multiple Organizations
  - Engineering
  - Product
  - Operations
  - Support
  - Different reporting structures to different VPs
- Virtual Team with Single Goal
  - All work to make Google App Engine successful
  - Coworkers are “Us”, not “Them”
  - When asked which teams we need to sit next to, it never occurred to us that other organizations were not “our team”

Why Are Organizations Slow?

- Organizational Culture
- Process
- People

Constant Learning

- Any process, organization, or product can always be improved
- Mistakes are a learning opportunity
  - What did you do -> What did you *learn*
  - Take emotion and personalization out of it
- Encourage iteration and velocity
  - "Failure is not falling down but refusing to get back up" – Theodore Roosevelt

Google
Blame-Free Post-Mortems

- Post-mortem After Every Incident
  - Document exactly what happened
  - What went right
  - What went wrong

- Open and Honest Discussion
  - What contributed to the incident?
  - What could we have done better?
  - ➔ Engineers compete to take personal responsibility (!)
  - ➔ "Finally we can fix that broken system" ☺

Google Blame-Free Post-Mortems

- Action Items
  - How will we change process, technology, documentation, etc.
  - How could we have automated the problems away?
  - How could we have diagnosed more quickly?
  - How could we have restored service more quickly?
- Follow up (!)

Iteration and Experimentation

- *Engineer* successes
  - Online products require constant iteration
  - Launch is only the first step
  - Assume you will not get it perfect on the first try
  - A / B Testing needs to be a core competence
- Many small experiments sum to big wins

eBay Machine-Learned Ranking

- Ranking function for search results
  - Which item should appear 1st, 10th, 100th, 1000th
  - Before: Small number of hand-tuned factors
  - Goal: Thousands of factors

- Experimentation Process
  - Predictive models: query->view, view->purchase, etc.
  - Hundreds of parallel A | B tests
  - Full year of steady, incremental improvements

➔ 2% increase in eBay revenue (~$120M)

Technical Tradeoffs

- Make Tradeoffs Explicit
  - Triangle: date vs. quality vs. features
  - When you choose date and features, you implicitly choose a level of quality
  - Be open and honest with yourself when you are doing this
- Manage Technical Debt
  - Plan for how and when you will pay it off
  - Maintain sustainable and well-understood level of debt

Why Are Organizations Slow?

- Organizational Culture
- Process
- People

Hire and Retain the Best

- Hire ‘A’ Players
  - In creative disciplines, top performers are 10x more productive (!)
- Confidence
  - A players bring A players
  - B players bring C players

Google Hiring

- Goal: Only hire top talent
  - "False Negatives" are OK – Google is willing to mistakenly reject a qualified candidate
  - "False Positives" are **not** OK – Google does not want to mistakenly hire an unqualified candidate
- Hiring Process
  - Famously challenging interviews
  - Very detailed interviewer feedback
  - Hiring committee decides whether to hire
  - Separately assign new Googler to group
- ➔ Highly talented and engaged employees

Respect People

- People are not interchangeable
  - Different skills, interests, capabilities
  - Create a Symphony, not a Factory
- Most valuable and irreplaceable asset
  - Treat people with care and respect
  - If the company values its people, people will provide value to the company

eBay "Train Seats"

- eBay's development process (circa 2006)
  - Design and estimate project
    ("Train Seat" == 2 engineer-weeks)
  - Assign engineers from common pool to implement tasks
  - Designer does not implement; implementers do not design

- ➔ Dysfunctional engineering culture
  - (-) Engineers treated as interchangeable "cogs"
  - (-) No regard for skill, interest, experience
  - (-) No pride of ownership in task implementation
  - (-) No long-term ownership of codebase

Recap: Why Are Organizations Slow?

- Organizational Culture
- Process
- People

Thank You!

- @randyshoup
- linkedin.com/in/randyshoup