Start-up Software Engineering

한기용 (keeyonghan@hotmail.com)

What I am going to talk about

- Brief Self Introduction
- Lessons from various companies I worked for
  - WiseNut
  - Inverito
  - Yahoo
  - Polyvore
  - Udemy
- Summary
- Important Skills
- Q&A

SELF INTRODUCTION

Brief Bio

- 21 years of industry experience as a software engineer/manager
- Worked on search and large scale data processing since 2000
  - Web Search, Hadoop based processing, ...
- Currently Director of Data at Udemy
  - In charge of Data Analytics, Data Infrastructure and Data Science

Work Experience

1. Samsung Electronics (Feb 1995 - Jun 2000)
2. **WiseNut** (Jun 2000 - Jan 2002)
3. **Inverito** (Feb 2002 - Apr 2003)
4. Looksmart (Apr 2003 – Feb 2004)
5. **Yahoo Search** (Feb 2004- Apr 2011)
6. Haileo (Apr 2011 – Dec 2011)
7. Hadoop consulting (Apr 2012 – Dec 2012)
8. **Polyvore** (Dec 2012 – Aug 2014)
9. **Udemy** (Aug 2014 - ?)

Jun 2000 to Jan 2002

LESSONS FROM WISENUT

What did WiseNut do?

- Founded in 1999 by Yeogirl Yoon in Santa Clara, CA
  - MySimon was sold to CNET at $700M
- Aimed to build another Web Search Engine
  - On top of MS Windows Server platform
- Got total fund of $11M and eventually sold to Looksm

What went well

- Built a very strong engineering team
  - Built 800M web search service from the scratch in 9 months using Windows NT servers
  - Most of them are now in Google or Bing Search
- Choosing “Web Search” turned out to be a very good decision
  - But we didn’t know at the time because of monetization issue (until Overture solved this problem)

What didn’t go well

- Leadership was mediocre at best
  - Tends to delay decision (small and big)
  - Built a bad relationship with VCs
- Bad timing
  - Acquisition discussion was actively going on with Overture and Altavista but 9/11 made all those discussions void
- Founding team was too close (inner circle)
  - No room for new comers

Feb 2002 to Apr 2004

LESSONS FROM INVERITO

What did Inverito do?

- Founded by 4 folks from WiseNut in Feb 2002
- Got an angel funding of $1M
- Aimed to build a SEC filing search service
- After a few pivots, launched an insider trading service called InsiderScoop
  - Reached 1,000 paid customers in 6 months
- Eventually the company went bankrupt in 2005

What went well

- Built two services in a short period of time
  - SEC filing search service called EdgarIQ
  - Insider trading service called InsiderScoop
- Insider trading service reached 1K paid users in 3 months
  - Subscription fee was $29.99 per month

What didn’t go well

- Turned out that market was too small
  - Our focus was more on technology side unfortunately
- Chemistry among 4 co-founders wasn’t great
  - Should have spent time knowing each other better
  - 4 is too many if they don’t have well defined roles
- SEC filing format changed to XBRL
  - Insider trading info used to be in HTML format but changed to XML from mid 2004

Feb 2004 to Apr 2011

LESSONS FROM YAHOO

What I learned from Yahoo

- Best Engineering Practices
  - Continuous Integration
  - Importance of unit test and successful build
  - Code review
- Importance of your professional network
  - Reputation matters
  - Very helpful (or hurtful) in job search
- How to build/manage engineering teams

- But I should have left Yahoo at least one year earlier

Yahoo Search Backend Team (Inktomi+Altavista)

- Smart people
- No or very little politics
- A great place to learn how to process large scale data (TB to PB scale)
- Well disciplined engineering culture
  - Focus on unit test and integration test, code-review and build success
  - 2 weeks release cycle
    - But didn’t use very strict agile development methodology

Why Yahoo didn’t do well?

- Hard to move fast
  - Process Centric
  - Matrix Organization
    - Engineering, Product Management, Operation, Science, QA
    - And from different geo-graphic locations!
  - Too many layers and politics
- Bad leadership
  - Couldn’t define its identity
    - Technology company or Media company?
  - Carol Bartz defined it as the latter but it didn’t go well
    - Outsourced major technology driven services (Web Search, Ads, Shopping Search) to Microsoft and other players

Dec 2012 – Aug 2014

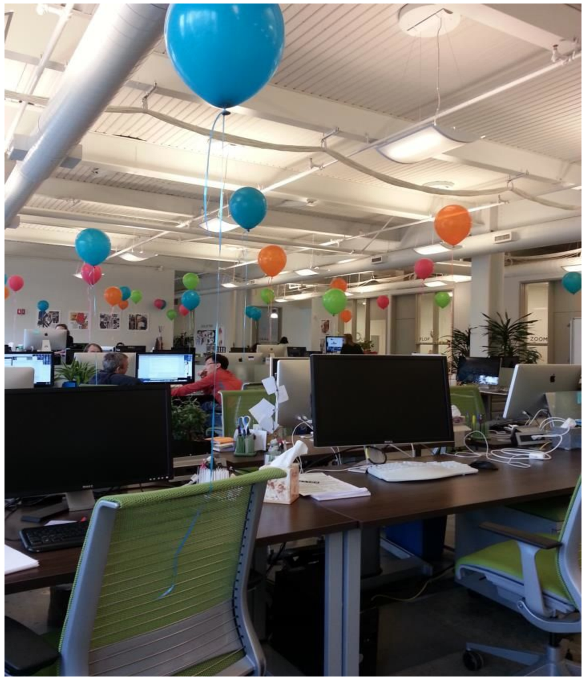
LESSONS FROM POLYVORE

What does Polyvore do? (I)

- **Polyvore is a new way to discover & shop for the things you love**
  - http://www.polyvore.com/
- Social E-commerce site focusing on women's fashion and home interior
- Users can create collages to express their own fashion or home decoration style
- Makes money from traffic referral
  - CPC/CPA/CPM

What does Polyvore do? (II)

- Founded by 3 engineers from Yahoo in 2007
  - Later a PM from Google maps team joined as the last co-founder (and now she is the CEO)
- Got total of $22.1M funding
- Now 100+ people
  - 60% are engineers/PMs/designers
  - 55% are females



THANKS, PASHA!
POLYVORE 2013

Lessons Learned from Polyvore (I)

- Learned a lot about B2C
- Move Fast (and Fail Fast)
  - A lot of A/B testing (metrics driven)
- Building **a strong community** matters
  - A word of mouth, most importantly SEO, …
  - Revenue follows user experience
- Unlearn old habits!
  - Have a mental break before joining any new company (or when you are promoted to a new position)

Lessons Learned from Polyvore (II)

- Continuous Integration to the extreme
  - Any engineer can push to production
  - On a busy day, there are 10+ production pushes
  - Yellow card/Red card!!
- Refactor/redesign your codebase before it is too late
  - **Balance between Speed vs. Technical debt**
- Importance of Hiring
  - Cultural fit matters
  - Define roles first
- Mobile changes Paradigm
  - Wanelo vs. Polyvore
  - Pinterest vs. Polyvore

Lessons Learned from Polyvore (III)

- **All kinds of growth pain**
  - Conflicts between co-founders and hired executives
  - Not easy to sustain original company culture
  - What to do with technical debt and old technology?
  - How to scale engineering organization as more people are added (from 15 to 50+)
- **Embrace changes**
  - The only constant is change
- Do fewer things well

Technology Stack

- Started with Perl and moving toward SoA
  - But migration isn’t easy after 7+ years
- Solr is used as the search engine
- Heavily using MySQL
  - Added Redshift for Analytics
  - Added Cassandra for certain access patterns
- Strict engineering discipline
  - Code review/Unit test is a must
  - Jenkins based CI is in place
- Scrum is strictly used (from Dec 2013)

(Paid) Services Used in Polyvore

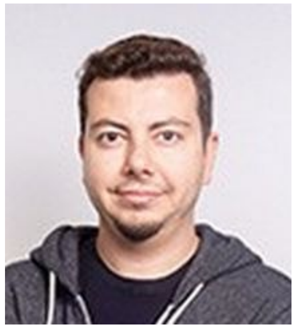
- AWS
  - S3, Redshift, CloudFront, …
- JIRA
  - Scrum/Bug tracking
  - SVN for code repository
  - Confluence for documentation
  - Crucible for code review
- github.com for code repository
- Jenkins for Continuous Build
- Slack for (group) chat/messenger
- Pagerduty for operation issue escalation
- JobVite for hiring lead tracking
  - Stypi or collabedit for coding question during Phone interview
- Expensify, Paylocity, … from HR perspective

Aug 2014 – ??

LESSONS FROM UDEMY

What does Udemy do?

- Founded by 2 Turkish folks in 2010 in SF
- Building a marketplace for online learning
  - 30K courses and a few Million users
  - Commission based Revenue model
- Total funding of $113M with 180+ employees



Lessons learned from Udemy

- Building a strong community matters again!
  - Instructor community and Student community
- Very effective email marketing and ads campaign
  - Analysts are really smart and knows how to leverage data
- Founders are smart enough to know their limit
- Being open is more important than good college degrees

Technology Stack

- Started with PHP and moving toward Django/AngularJS
- Heavily using MySQL
  - Added Redshift for Analytics
  - Added Cassandra for certain access patterns
- ElasticSearch is used as the Search engine
- Relatively loose engineering discipline
  - Code review is a must and CI is in place
  - But unit test is not a must in check-in
- Scrum is used in terms of planning and daily stand-up but not strict

(Paid) Services Used

- AWS
  - S3, Redshift, CloudFront, Video Encoding, …
- Trello
  - Scrum/Bug tracking
- github.com
  - Code repository, Documentation and Code Review
- Jenkins for Continuous Build
- HipChat for (group) chat/messenger
- Pagerduty for operation issue escalation
- DataDog for monitoring
- Expensify, Jobvite, …

SUMMARY

My 2 cents: What to Remember

- Co-founder(s) and founding member(s) matter
- Don’t try to do too many things (features)
  - Specialize rather than Generalize
- Build your community
  - Focus on user experience (delight users)
- Mobile first
- Size your market from business perspective
  - Don’t look at things from technology

Engineering Side – Technology Choice

- Speed is critical in the beginning
- Technology choice is important
  - Something you are familiar vs. something the most relevant to what you are building
    - Certainly you don't want to use Perl ☺
    - MySQL can be used for everything but …
  - This will affect hiring as well
- Important thing is to be flexible and open-minded
  - SoA architecture can be helpful

Engineering Side
AWS or My own servers?

- AWS is good to start
- But it is not cheap if you are not careful

1. Monitor AWS usage
   - ICE is a good tool to install from Netflix
2. Make sure to turn off unused EC2 instances
   - Use EBS backed instances (but it is slower)
3. Network bandwidth can be costly
4. If possible, make your software elastic
   - Leverage Auto-Scaling feature
   - Automation helps as well

Engineering Side – Code/Build/Test

- Github is now a de-facto standard
- Jenkins based CI from the beginning can be really helpful
  - Have some basic structure for unit-test and integration test
  - Onboarding a new engineer will be a lot less hassle once you have this

IMPORTANT SKILLS

Basic Useful Skills

- Python
- Github
- SQL

- Most importantly capacity is more important than skill
- Only Constant is Change

Q & A