What You Need to Unlearn from School and Other Startup Lessons

Presented by Steve Kaufer, CEO of TripAdvisor

Hosted by Professor Harry Lewis, SEAS, and Harvard Computer Society

April 18, 2013

Introduction

Part 1: Setting the context

Part 2: Unlearning a few lessons taught at school

- Proving your Thesis
- The Term Paper

Part 3: A few lessons learned the hard way

- Engineering
- Hiring
- Luck
- Getting stuff done

Part 1: Setting the Context

Original company, CenterLine did not adapt to release of Windows. Stated to decline because couldn't figure out how to do things differently.

TripAdvisor - Wanted to use a vertical search engine for travel. Something to type into that was better than AltaVista. Great candid advice and photos about places to travel. Launched TripAdvisor in 2000.

Part 2: Unlearning lessons

Prove your thesis

School - teaches you develop thesis, then support it with facts and persuasive arguments
That's exactly what you do to start a company, you need to raise money.

- Thesis - need you are going to meet
- Facts - explain why the market is huge, the need is great, the customers will want what you have
- Persuasive arguments - why your team is the right group to address this problem. Why are you going to make this a successful company?

Next round, you have to show that this product is something wants to invest in. In TripAdvisors case, before they had any "proof" that their product would succeed, they raised $2M. Pocketed money on "hopes and promises" By 2001 they had run out of money. If they hadn't got $2M early on, with no proof of success, they would have failed early on. If they tried to get money a year into it, investors would have asked to see success, and they didn't have any. **Moral: Get money early on.**

College Term Paper

University Method

- Due in 4 weeks - No credit for handing it in early. Huge penalty for a day late.
- Major penalty for typos - details in footnotes are as important as paper
- Work alone - you can't have people writing your paper for you
- Copying - don't reuse previous work (plagiarism)

How does this method apply to the business world?

- **Deliver it early.** Do our customers want 80% solution and start working on the next feature, or 100% solution
- Who cares about a typo? Don't stress about the details. "**Perfection is the enemy of productivity**" We need to keep iterating. We're always in a hurry!

- Work in teams. Teams deliver better results. **If you can get someone else to do your job: awesome!**
- Please reuse previous work. Not only is it faster, there is less to maintain.
    - Whether it's past work from inside the company
    - Copying exactly what another company does if it does well
    - Open source
    - Industry best practices
    - Alright to have original thought to improve something that already works well
    - learn from others; use cornerstones of other companies to be your design guide

A Few Lessons Learned the Hard Way

Lesson #1: Running an Engineering Department

Testing Your Code

Biggest challenges - How development related to QA, and how it usually didn't come together. Who has what responsibility?

- So who really does QA? Developers or QA staff?
    - This is a personal choice
    - In TripAdvisor it's the developers. They should be responsible for product end to end.

When developers are adding or changing something, they are 100% responsible for something that they change. It's not alright to have another part of the system mess up.

- What type of bugs are "ok?"
    - The minor ones on the new feature.
- What type of bugs are "not ok"?
    - Ones that cause regressions on the old one.
    - QA guys at TripAdvisor are responsible for making sure that new features don't break across multiple browsers or in environments not easily accessible to feature developers.
- What happens if a developer really screws up?
    - No finger pointer.
    - A lot of pressure to not mess up, so people usually don't
    - If it's a repeated offense:

Speed

Weekly release cycles at TripAdvisor, across 20 languages, 30 websites and dozens of features/fixes going live each week. 3 QA.

Software is getting more complicated, more languages.

Perfectly happy to support whatever engineering team needs to keep fast cycle. It is VERY important to keep a fast release cycle.

"**Reward the behavior you are looking for..**."
Back in the day, when Steve was engineering manager of his old company, the release cycle was really slow. Things were off schedule. 7 month expected release happened in 9 months. This happens ALWAYS. His boss gave him only half of bonus for quarter. Seriously contemplate whether you should ever be late again.

Don't ever be late! Next time when he specced it out, he figured a 6 month schedule but told heads, 11 months. They got a *reliable* engineering schedule, not a *fast* engineering schedule. That's what they wanted.

Now at TripAdvisor when things aren't going according to schedule, that's fine. This doesn't work when a company has big external commitments. Aim for speed with acceptable quality. Don't get mad when things come in late.

Lesson #2: Hiring

Interviewing is a skill - learn how to do it well.

Learn how to ask good questions.

“A”s hire “A”s and “B”s. “B”s hire “B”s and “C”s.

- Why don’t B’s hire A’s? B’s have nervous anxiety; afraid of being “showed up”

Test communication skills. Can they communicate something they know about their topic?

Better to be safe than sorry. CEO saves company a lot by staying close to hiring process.

At TripAdvisor

- Don’t close open headcount
- Don’t have deadlines to hire people
    - “You need to spend more of your time looking for this person”
        - Allocate resources, but don’t demand deadlines
- Don’t care about recruiting “costs”
- People are the most important.
- We recognize that a ‘star’ is 2X more ‘valuable’ than strong and ‘strong’ is 2X more valuable than ‘average’
- We manage people out throughout the year, but never fast enough.
    - Is your company a ‘family’ or a ‘team’?

Mid-presentation announcement: We’re hiring!

Lesson #3: Luck and timing is important to recognize

“The harder I work, the luckier I get.”

- Don’t confuse luck with brilliance.
    - TripAdvisor had great timing.

At TripAdvisor

- “With the benefit of wisdom, but not hindsight, what should we have done differently?”
- Did I learn anything from process that would allow me to do anything differently
    - Great way to go through self improvement

Lesson #4: Random tips about getting stuff done

Internal motto: “Speed Wins”

Be willing to fail fast
“How can we do this in half the time?”
“3 months or 3 days?”

“If it is worth doing, it is worth measuring.”

A facebook app appeared that tracked places people have been. Was hugely popular and Steve wanted it for TripAdvisor. Engineering head gave a 3 month estimate. This was too much time, they would have lost the social travel space to the app. All resources were allocated to make it 3 days.

Lots of debates are settled by saying: “both opinions are valid; test it.”

- subject line of emails, placement/color of button, tv ad concepts. (A/B testing)
- But what about a new feature? 404 tests

Example: 404 Testing on Personalized List Feature

Put up a 404 test, ask you those few questions. “give you a recommendation or no thanks.”

Here’s how: We show a popup asking if you would like a personalized experience. You can say yes or no.

It takes 1 day to write this popup. If you say yes, just tell the user “404, sorry, service is not available!”

Now we can see who is interested and who is not, but we only spend 1 day making this popup. If its not liked, **you haven’t wasted much time**, and now you don’t build out the feature! If it’s clear that people want to try it, then it’s clear that you should **proceed** with the project, not kill it.

Real world feedback is important. Don’t underestimate it’s use!

Steve still does Q/A.

Question and Answers

Q: You need money, but people don’t understand your idea. What do I do?
A: Expect 19 no’s before somebody says, “I kind of understand what you’re saying” If you know why you got a “no” you can fix up your question so people do get t. Target angel investors and VC’s that know hte business.

Q: Are you worried about getting sued when you “borrow” ideas from other companies?
A: Not explicitly stealing. Stealing “inspiration”. Taking how users interact with the site.

Q: Model of TripAdvisor is “faster, faster, faster” are you talking about development cycle or number crunching? Is processing data in real time advantageous?
A: When a user comes back to the site, they instantly have a personalized site based on what they did on Facebook. We don’t have the power to do that yet. Our “speed wins” motto applies to how employees can become more efficient.

Q: Did you buy “Where I’ve Been?”
A: NY Times said so, it was false reporting. We bought it 5 years later. We couldn’t agree on a price at first because of MySpace. Later, when they went on sale, we bought it as a defensive

measure.

Q: What’s the threshold of the 404 test to determine success?
A: Depends on the test itself. For personalized recommendations, 10% is great! People will give it a try, so I build it. 5% might start telling their friends. 5% on TripAdvisor is millions of people. Has ability to have scale.

Q: How does the “move fast” mentality move to other spaces?
A: We blessed to be in the internet space. If we try something and piss people off, we can still revert and life goes on like normal. The cost of failure is very low. For NASA, x-ray machines, radar systems etc., this is not feasible. Embedded software for things like x-rays, NASA, radar, etc have too high of a cost of failure that the business model that is so suitable for the internet space can’t be used in the same way. Being faster and relying on speed works in the internet space but it won’t work when you’re deploying the next x-ray or radar.