2012 Agile Conference

Applying Lean Startup: An Experience Report

Lean & Lean UX by a UX Veteran: Lessons Learned in Creating & Launching a Complex Consumer App

Beverly May
Founder & CEO, MiniDates, Oxford Technology Ventures, LLC and the User Experience Awards
New York, NY
Beverly.May@OxfordTechnologyVentures.com, @OxfordTech, @MiniDates

Abstract- *This paper outlines lessons learned from a senior UX, product and software development consultant's experiences in conceiving, designing, developing and launching MiniDates.com, a complex consumer dating application. Key decisions and moments over our 15 month process are highlighted, including what we worked and what didn't, what we should have done that we didn't, and what we encountered that we couldn't have foreseen, in the belief that we can help other UX and product-focused startups and Agile teams be more effective. Key lessons include: 1. Beware fads and advice from others; 2. Optimize your team and the development process as much as possible; 3. Beware of seemingly innocent technological and architecture decisions; 4. Ensure UX planning, design and customer validation are up-front in your process; 5. TEST, TEST and TEST some more- at all levels of the business and at all stages, not just the software itself.*

Keywords- Lean, Lean UX, Agile, Case Studies, Customer Development, prototyping, bootsrapping, startup, customer acquisition, entrepreneurship, testing, usability, research

I. Introduction

> *"We thought we should get a gold star just for listening to our customers. Except our customers still didn't like the product. They would look at us and say, 'Listen, old man, you don't understand...'" [1].*

A. *Do as I Write, Not as I Did!*

The above quote is my favorite part of Eric Reis' *Lean Startup* and the inspiration for this paper. It encapsulates the optimism, arrogance and foolishness of traditional software development, and it accurately represented my experience over much of my 16 years in software and the Internet in helping build and launch dozens of ultimately failed or short-lived products. One reason *Lean Startup* is so influential is because Ries was not afraid to share both his mistakes and his successes.

This paper is also a "lessons learned" confessional of our experience creating, launching and marketing a sophisticated new consumer dating application called MiniDates over the last 15 months. The summary? Do as I write, not as I did! I am sharing our experiences with you, dear Agile reader, as I believe there are many lessons to be learned from both our successes and our failures.

II. Background

We were not new to software development when we began MiniDates; I began in the Internet in 1996 and have over 16 years' experience in helping concept, design, develop, manage, launch and optimize well over 100 products and services. I've worked on truly global, enterprise software for the UN, spent a few years in the top digital agencies, and have been a part of several startups, including one that was sold for over $400 million. I have an MS in systems design from NYU, and an MBA from Oxford University. I'm President of NYC-CHI, New York City's largest and oldest UX organization. The original application architect also had about 10 years' development experience developing sophisticated enterprise software and mobile applications.

Most importantly, our consultancy, Oxford Technology Ventures LLC, specializes in helping others launch successful products and services in the market through User Experience, Usability and Product Strategy services. This includes concepting, wireframes, prototypes, personae, requirements, design, user research, user testing, funnel and conversion flow optimization, navigation, content hierarchies and content strategy, and much more. We and we have a range of clients from early-stage startups to Fortune 100 companies. Our three principles are to assist our clients in delivering products that are 1. Effective for the business, 2. Useful and usable for the user, and 3. Engaging-prompting stickiness, delight and repeat visits. We're also the creators of the User Experience Awards. Given all this, we felt we knew something about how to create and launch a successful user-centered product!

A. *MiniDates!*

MiniDates.com was originally conceived as a showcase case study for our consulting business. The idea was to create a

978-0-7695-4804-3/12 $26.00 © 2012 IEEE
DOI 10.1109/Agile.2012.18

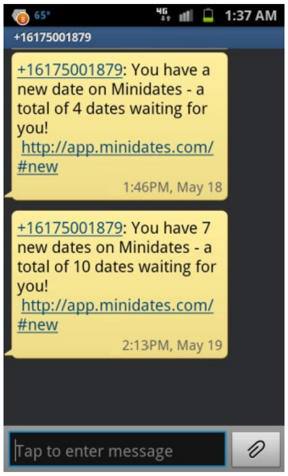
superior, significantly differentiated customer experience, using a cutting edge mobile tech stack and Lean and Agile development processes. This was to be our crowning glory to show we could "walk the walk" in designing, developing and launching a new product, soup to nuts, rather than simply giving direction and advice to others as outside consultants.

What is MiniDates? It's what we call the first "real" dating service that sends you planned, in-person date proposals by text, email or in-app notification based on both who you're seeking and when and where you're free. We're the first service to match based on compatibility, schedule and location, and we're the first to send members actual planned dates (vs. lists of user profiles).

Traditional dating services have barely evolved in the past 15 years. Most services provide you with a search or browse experience of profiles and leave all the work to members to first find the right person, then message, and then plan dates, and usually there's a lot of drop-off and frustration that occurs because the process is inherently inefficient, full of rejection and time-consuming. Minidates does the opposite- we send members actual dates up-front for them to accept or decline. MiniDates is easy, fast, fun, convenient and *efficient.*

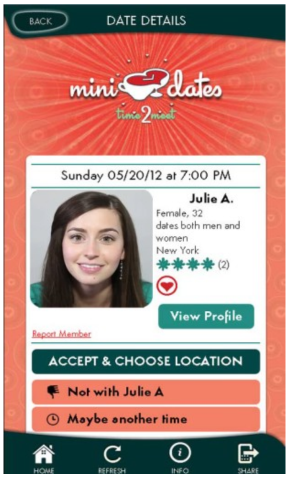
MiniDates' key differentiator, clearly, was in the product user experience -- and the underlying technology supporting that experience. We decided early on to build MiniDates as an HTML5 mobile web app for universal platform accessibility and a consistent interaction experience.

Of course, once MiniDates was begun, it soon took over. It became THE project for Oxford Technology Ventures, not one of many. In 2012, the company basically pivoted to become a full-blown startup, focusing about 90% of our resources and time on MiniDates.



+16175001879: You have a new date on Minidates - a total of 4 dates waiting for you! http://app.minidates.com/#new
1:46PM, May 18

+16175001879: You have 7 new dates on Minidates - a total of 10 dates waiting for you! http://app.minidates.com/#new
2:13PM, May 19



Sunday 05/20/12 at 7:00 PM

Julie A.
Female, 32
dates both men and women
New York
★★★★ (2)

Figure 1. (Left) MiniDates SMS Notifications of new dates

Figure 2. (Right) A sample date proposal on MiniDates

B. *Definitions*

What do we mean when we speak of "Agile", "UX", or "Lean" in this paper?

1) Agile-

Let's start with the basic tenets of the Agile Manifesto, e.g., individuals over processes and tools; working software over documentation; collaboration over negotiation; and responding to change over following a plan. In summary: iterative, nimble, product-focused team development. Add in 'best practices' of Scrums, Sprints, User Stories, short release cycles and frequent releases, test-driven development or pair programming, and dedicated cross-functional teams [2, 3, 4].

2) UX-

User Experience best practices focus on user-centric product concepting and design (UCD). This means designing and architecting the product from the user's experience, based on her needs, goals, and the product context. The product's flow and functionality is defined and designed around fulfilling her needs through an optimal end-end experience. This encompasses: creating user stories; personas; card sorts; user research (ethnography); wireframes; user flows; prototypes; sitemaps; and testing and validating all research, concepts and prototypes with the end users [5, 6, 7, 8].

3) Lean-

Based on the work of both Eric Ries and Steve Blank, the "Lean" applies Agile and UX best practices and approaches to the business [9, 10, 11]. Lean encompasses: searching for and validating a viable, sustainable and repeatable business model; performing customer development by 'getting out of the building' (GOOB); creating products customers will use and pay for; validating the business type and growth hypotheses, amongst other hypotheses, through the build > measure > learn feedback loop; creating an MVP and possibly a "Concierge" MVP; emphasizing resource efficiency in learning; and directly validating the business by the business leads themselves, rather than by consultants.

III. Chutes & Ladders: Product Evolution & Timing

Chutes and Ladders was one of my favorite board games as a young child [12]. It is a metaphor for the ups and downs in life wherein on rolling dice, one either skips many steps ahead on the board by climbing a ladder, or reverts many steps back by going down a chute. Chutes and Ladders is an apt metaphor for a startup and how difficult is to make real progress.

To describe our evolution for this paper for Agile 2012, I created the MiniDates version of a "Chutes & Ladders" diagram (though ours has neither ladders, nor chutes). What the large **diagram in Appendix A** shows is a snaking timeline of all the major milestones and events that happened during our concepting, development and launch, color-coded by milestone or event type.

A. Key MiniDates Milestones (Black dots)

We embarked on developing MiniDates in late January 2011 after a few early discussions and concepting, and ultimately launched the service in beta 15 months later- when we had originally thought it would take us only 3-4 months to functioning alpha and maybe 4-5 months to live product!

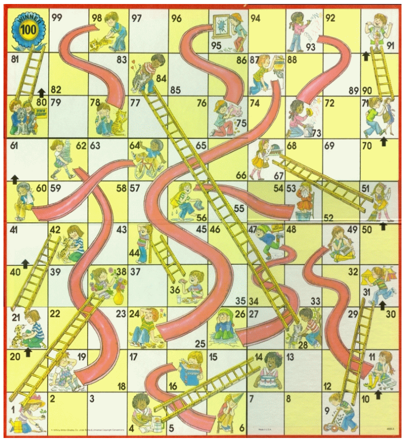


Figure 3. Chutes and Ladders, circa 1977
© Milton Bradley and Hasbro.

Why did it take so long, especially when we had such extensive experience? Aside from the old adage that things always take at least twice as long as you can ever imagine they might, there were five other reasons of note:

1) Conflicting Demands on Time & Attention:

For the first 8-9 months of the project, we were very busy with major client projects, leaving the software architect to basically fly solo without much direction or oversight. The consulting was necessary to pay for the development and keep the lights on, but it also slowed down the product, design and customer development process considerably (see the double red and black line in the diagram). It wasn't an ideal situation, but it was necessary.

2) Early Technology Architecture Choices:

We made some early technical decisions that turned out to really slow development and significantly increased technical complexity, particularly the choice to use HTML5 rather than create native apps. I can confidently state that if we had simply made an iPhone app, we could have been launched in our original timeframe of 4-6 months. Going the HTML5 route literally tripled our time to market, and our costs- to little benefit (see green dots in the diagram).

3) Overly hasty UX and Design Decisions:

You'd think this is one process element that we would have aced, but we foolishly thought that because we WERE UX experts, we could cut corners and skip a few steps by going straight into visual design in some cases (skipping wires, prototypes, and/or more detailed specs/logic flows). It turns out that was as bad an idea as a developer trying to skip QA, and inevitably cost us more time and cycles in iterations and modifications (see light blue dots in the diagram).

4) Too Little Customer Development and User Research/ Market Testing:

Too little, far too late. Until I read Eric Ries' book in October 2011, I looked at customer research as an early step in the UX-related user-centered product design process. His book was a revelation that customer development could and should be approached with the aim of validating the *business* assumptions and particularly gauging demand and refining product –market fit and positioning *before* building a product. We embarked on this far too late.

As well, the results we received were mixed, inconclusive and murky. We made a choice to plod on regardless- after all, this was our baby, and we wanted to see it live! What actionable results we did receive from these tests contributed to leadership insecurity and a delayed product launch by introducing new product complexity: new features, new names, new designs, and an early customer acquisition campaign that was ill-conceived and executed (see red dots in the diagram for product/business testing and yellow for customer development and marketing).

Doing Lean with a startup is fraught with difficult philosophical questions: what to do if feedback is negative? How to get the information you need, inexpensively and quickly? How much validation is sufficient? As Eric Ries notes, almost all products have some degree of success and some users, with some redeeming value- but *some* is not usually enough to generate success [13]. The key learning is that Lean exposes the inherent underlying business risks upfront, like going through an airport X-Ray machine, and that can present very uncomfortable and difficult decisions.

5) Poor and Unlucky Team Hiring and Management:

This was, by far, the biggest factor of all. In total, the app has gone through 5 Tech Leads in 15 months. Going back to the technology architecture choices that were made initially, our architect's selection of a cutting edge stack (MongoDB, Python, Tornado, NGINX, lots of Javascript and related libraries, and HTML5) meant that this project's potential pool of talented architects who were familiar with all these technologies was small, and those individuals were in extremely high demand. Had we used even slightly more widely-adopted and established technologies, such as iOS native development, this would not have been the huge problem that it became. Couple that with massively unlucky hiring and insufficient day-day managerial oversight of the development until the last few months of the project, and it was a recipe for disaster.

Suffice to say, everything that could go wrong with the development, did, but our hiring woes were also not limited to technology. We also engaged ill-fitting and low-productivity associates as well as sales, product and marketing managers that reduced overall morale and productivity and drained our limited resources without much output. It turns out it's really hard to find talented, self-motivated, experienced people who have a DIY ethos and want to work on an unfunded startup, even if you're paying (see purple dots on the diagram).

B. Agile, UX & Lean

Take note: we often did not follow Agile or Lean (or UX) "Best Practices" with MiniDates. Why, when this was one of our stated goals? Sometimes we were trying to cut corners with the aim of getting the project launched, and sometimes the approaches did not appear to fit the project or the team structure, and sometimes we were just plain foolish (and learned the hard way).

Agile: We had one architect working solo on the project for about 2/3 the project; when others joined, they were as designers or QA. No team= no "team" Agile practices. Until we were far enough along to have a functioning alpha, defining strict "sprints" for a team of 1 seemed like project management overkill, and even when those definitions were created, they were inevitably missed, or misinterpreted. As well, what's "done" for an MVP, or even an alpha, and especially from a developer's perspective, is rarely likely to be sufficiently polished or "complete" for a customer-facing beta (it would be interesting to read Agile best practices for startup dev. teams of 1 at the earlier stages of a project, but that is not the focus of this paper).

We started implementing Agile methods near the end, on Tech Lead #4 and #5. We now have daily scrums, and we release once a week (see orange dots on the diagram to follow our development process evolution).

UX: Given our specialty in this area, we should have gotten this right, but as noted previously, we tried to cut corners here to save some time and money. There are many things we did 'right': early wires of some views, using a UCD design process, personas and wires, and plenty of market and user research. Things we skipped: rapid prototyping, maintaining the wireframes and completely diagramming the app and all flows based on HTML5 standards, and early and frequent user testing of all parts of the app and its demand (see teal dots on the diagram for the UX and design process).

Lean: We did search for a viable, revenue-generating business model and did significant GOOB research on pricing, features and motivators. We began customer development in earnest by signing up users to use the service well prior to launch, starting in January 2012. We didn't do a real MVP, particularly a "concierge" MVP that could have validated the business model and workflow without writing any code, and we didn't validate the business and the revenue and economic assumptions nearly early enough [14] (see red dots on the diagram).

C. Where We Are Now

We finally have a launched product. It works pretty well, and is stable most of the time. It looks great, and is totally different than anything else out there. It does what it says, and some people immediately get the value. We had great reception at SXSW, at NYC Internet Week, and at NYC startup pitching events. By launch, we had 3000 people on our customer acquisition waiting list, most of whom we'd gotten through GOOB: actually talking to people about the product and getting signups through a street team of "brand ambassadors" that went into bars and clubs and talked to singles. Sounds like great early success, right?

Yes, except for one problem: we don't have users. All those customer development names we got are not converting, and the ones that are, are converting at a much, much lower rate than expected. Publicity is slow and hard to come by. We're out of money and are finding it harder and slower than we ever imagined to raise funds, as well as to convince new customers to trying the service. We're at exactly the chasm that *Lean Startup* was supposed to help us avoid- a launched product in search of a customer base and a viable business model. We think we know our target customer base- but we're having trouble reaching them effectively, and they're a very expensive group to reach. The average customer acquisition cost for a paying customer in online dating is over $50/user. We can't afford that without funding, which we don't have. What will we do?

True answer: we don't know. We're going to give it our best shot to grow and refine our marketing methods and conversion funnel this summer, get some press and get out there. Truthfully, this is 90% the challenge of any startup, but since most of us come from product and tech, we don't think about how unbelievably hard and essential the marketing and customer acquisition part is. Ideas on paper, and even early traction or perceived validation, are just not sufficient. You are trying to create a business, not a product. The product is in support of the business, not vice versa. It's easy to say and very hard to implement.

IV. Top 5 Recommendations

So what can you, dear startup reader, learn from all this? Here are our top 5 recommendations:

1. **Beware fads & advice from others**
2. **Good UX IS Essential**
3. **Choose your technologies carefully**
4. **Optimize your team and the development process**
5. **Test EVERYTHING, early and often**

1) Beware fads & advice from others (including Lean & Agile!)

Two weeks before our launch, I interviewed a prospective CTO. At the contract negotiation phase, he told me, "Yours is a 'death march' project. You will *Never Launch*!". Two weeks later, we launched.

The lesson? People will give you all kinds of advice, and a lot of the time, it's inaccurate, misguided, inappropriate for your project or just plain bad. You have to stand your ground and maintain a clear direction and point of view of your own while continuing to be open to others' suggestions and advice.

There are countless cases of advice we received that was eminently *un*helpful. Some examples:

- **Lean:** Lean Startup Machine and others will tell you to go do some SEM marketing and send some users to your landing page to test interest and messaging. Yeah, we tried that. We blew through $1000 in Facebook and Google Adwords in a few days, and learned pretty much nothing at $3.50/click (dating is one of the most expensive SEM areas there is). Turns out, Adwords don't cost 5 cents any more like they did for Eric Ries in his book- you can burn through a lot of money very quickly using this tactic. Then, we got flagged by Google for "information harvesting" because neither Lean Startup Machine, nor Unbounce, nor Launchrock had advised us to put a clear Privacy Policy and opt-out process on our lead-gen landing page.
- **Staffing:** I loved hearing from various accelerators and others that we just HAD to have a CTO cofounder. Guess what? We launched just fine, thank you, but we wasted many months chasing these mythical tech Unicorns when we could have been working on launching and growing the business.
- **Fundraising:** I can't tell you how many times I've been told, "why not fund MiniDates through Kickstarter?" Yeah, that WOULD be great, except Kickstarter doesn't support commercial software and services (only commercial physical products, and movies, and everything else you can think of). As of this writing, there is no legal, crowdsourced option for funding commercial software (yet).

You get the idea. By the time you get advice, it's usually too late, and it's often inaccurate.

2) *Good UX IS Essential.*

Ensure UX planning, design and customer validation are up-front in your process. Don't do what we did and try to skip wireframing and prototyping, and be sure to test your wires, prototypes and designs on real users before building them. Be ready for brutally negative feedback at your first user tests, even from friends and sophisticated techies that should "know better". You'll likely hear what you already know in terms of what needs to be fixed, but it's good to be validated about that. Also, make sure you test your names! We've gone through 4 names for our consumer service.

3) *Choose your technologies carefully*

This was touched upon already: HTML5 and our cutting-edge tech stack added about 3x cost and time to market to our project. Be sure you know what you're getting into when making critical architecture and technology choices up-front, and try to make choices that are suited to the phase of the project. For most startups, **fast, nimble/flexible and inexpensive** should be the 3 criteria for evaluation of any technology solution or approach. Our chosen stack and approach was slow (HTML5 is complex), non-flexible (because resources were too rare) and ultimately very expensive. If you're the business lead, ensure you understand the key technology platform decisions that are being made by the tech leads, as well as the expected costs and their reasoning.

Also, resist most developers' desires to work with the latest hot stuff and introduce "technology bloat" just because some new framework, library or language is cool- that's like driving a Ferrari to the grocery store when a bike would have worked just as well. We ended up stripping a lot of unneeded technological complexity from the original architecture (SASS & Coffeescript) before launch.

4) *Optimize Your Team & the Development Process*

I recently went to a VC talk in which the VC listed the top 10 reasons startups fail [15]. Every single reason had to do with the team. Hiring is one of the hardest tasks for startups. Seriously consider where you're based, and how you will acquire key staff and partners. This is an area we have repeatedly failed at. Always be on the lookout for possible team members, co-founders or partners.

Equally as importantly, don't be afraid to **fire fast**. Underperformers are not just B or C-level in productivity; they drag everyone else down, require a lot of oversight, and they have a signaling effect which announces, "we are a b or c level company that rewards mediocrity". You MUST be able to trust each of your team–members to do his/her job fully, completely, very well and independently. You may think you'll be worse off without a mediocre performer in a key role, but 99% of the time, you'll be much happier when you let them go.

On the process side, daily Scrum is probably the most valuable process we instituted to help manage and monitor team performance. Also, don't be afraid to diverge from Agile (or any other) dogma and iterate and adjust your process to suit your team and the personalities that are on it, as developers need and want varying levels of oversight, involvement, specifications, QA, collaboration and team engagement.

5) *Test Everything- early and often.*

Testing here refers to "trying, evaluating and/or researching". QA testing and user testing are only two forms of testing. You really need to test your product fully and completely from the very first moment, ongoing. First you test the concept with some sample users. You test who your audience is, what their motivators are, what their pain points are, your concepts, your prototypes, your personas, your business projections, your wireframes, your designs, your alpha/ MVP, your beta, your marketing, your message, your branding, your positioning, your competitors, and your assumptions. Testing is not a tool for one step of the process. Testing is a metaphor for the entire process of creating a startup and refining your approach across the board, at every step.

The summary? TEST, TEST and TEST some more- at every step, and at all levels. Test the advice you're given, the UX and product direction, your technical decisions, your hiring decisions. As well, the key lesson of Lean is that the most important things to test are related to the business, not tech or product. You need to test your business assumptions; your metrics; your drivers; your model; your idea of your target customers; their motivators and needs; their willingness to pay; who they are and how you will reach them; how much it will cost to acquire them; how long the sales cycle is; and how others have done it successfully.

V. Conclusion: PLEASE TEST

Launching a startup is still really, really hard. Yes, it's easier than ever before, but that doesn't mean it's easy. An oft-cited stat is that 1 in 1000 startups raise funds, and 1 in 100 of those make it- so you have a 1 in 100,000 chance [16]. In particular, trying to do what we did was particularly challenging (and ultimately ill-conceived). We chose to:

- Launch a startup in a highly competitive, saturated and expensive industry. There are 2000 dating sites in the US alone (and 99.9% are almost identical from a product experience perspective). Match.com made $520M in gross revenue in 2011, and spent a massive chunk of that on advertising. It's hard to compete against that! And before you suggest that we're simply segmenting a mature market, remember that segmentation is still competition.
- Launch a new type of sophisticated, proprietary technology in an industry where technological sophistication has not been historically valued, using cutting edge languages and frameworks that were not yet well adopted and expensive to hire developers for.
- Launch a startup while trying to run a consultancy. Too little time means too slow to market.
- Launch a startup with only 1 true founder and no consistent CTO. The dogma is that you need a founding team- successful startups rarely have one owner/ founder.
- Launch a startup with only internal funding and no prior experience in raising outside investment. Fundraising requires a strategy of its own that is not apparent from reading the latest *TechCrunch*. We simply had no idea how difficult it was going to be.

The odds were, and are, overwhelmingly stacked against us. Add into that all the unknowns, risks, costs, people challenges, personal challenges and complexity of any startup, and it's a wonder that any project makes it.

The key thing to note? **Almost all of these challenges could have been uncovered much earlier, and strategized against, by testing the business assumptions and feasibility of the business up-front.** More than likely, it would have had us rethink our process or approach, or drop the project altogether.

I believe that aside from experience and smarts (which are still valuable), the **only** things that can help ensure success are extraordinary passion, effort, perseverance, luck, **and a lot of fast learning and smart decision-making**. **TESTING** is truly the keyword. I hope that this paper will be helpful to other startups and product teams in assisting in their success by adopting this ultimately simple "test" concept.

References

[1] Ries, Eric. The Lean Startup. New York: Crown Business, 2011, p. 16.

[2] Appelo, Jurgen. Management 3.0: Leading Agile Developers, Developing Agile Leaders. Boston, MA: Pearson Education, 2011.

[3] Cohn, Mike. Succeeding with Agile: Software Development using Scrum. Boston, MA: Pearson Education, 2010.

[4] Rasmusson, Jonathan. The Agile Samurai: How Agile Masters Deliver Great Software. Raleigh, NC: The Pragmatic Bookshelf, 2010.

[5] Garrett, Jesse James. The Elements of User Experience. New Riders Press, 2nd Ed., 2010.

[6] "Human- Centred Design Processes for Interactive Systems". Retrieved from: http://www.iso.org/iso/catalogue_detail.htm?csnumber=21197 , International Organization for Standardization, 2011.

[7] "What is User Centered Design", Retrieved from: http://www.upassoc.org/usability_resources/about_usability/what_is_ucd.html , Usability Professionals' Association, May 2012.

[8] "User Centered Design", Retrieved from: http://en.wikipedia.org/wiki/User-centered_design , Wikipedia, May 2012.

[9] Ries, Eric. The Lean Startup. New York: Crown Business, 2011.

[10] Blank, Steve. Four Steps to the Epiphany. CafePress.com, 2005.

[11] Blank, Steve. The Startup Owner's Manual. K&S Ranch, 2012.

[12] Chutes and Ladders. http://www.hasbro.com/shop/details.cfm?R=8EC0A8E0-6D40-1014-8BF0-9EFBF894F9D4:en_US Milton Bradley and Hasbro, May 2012

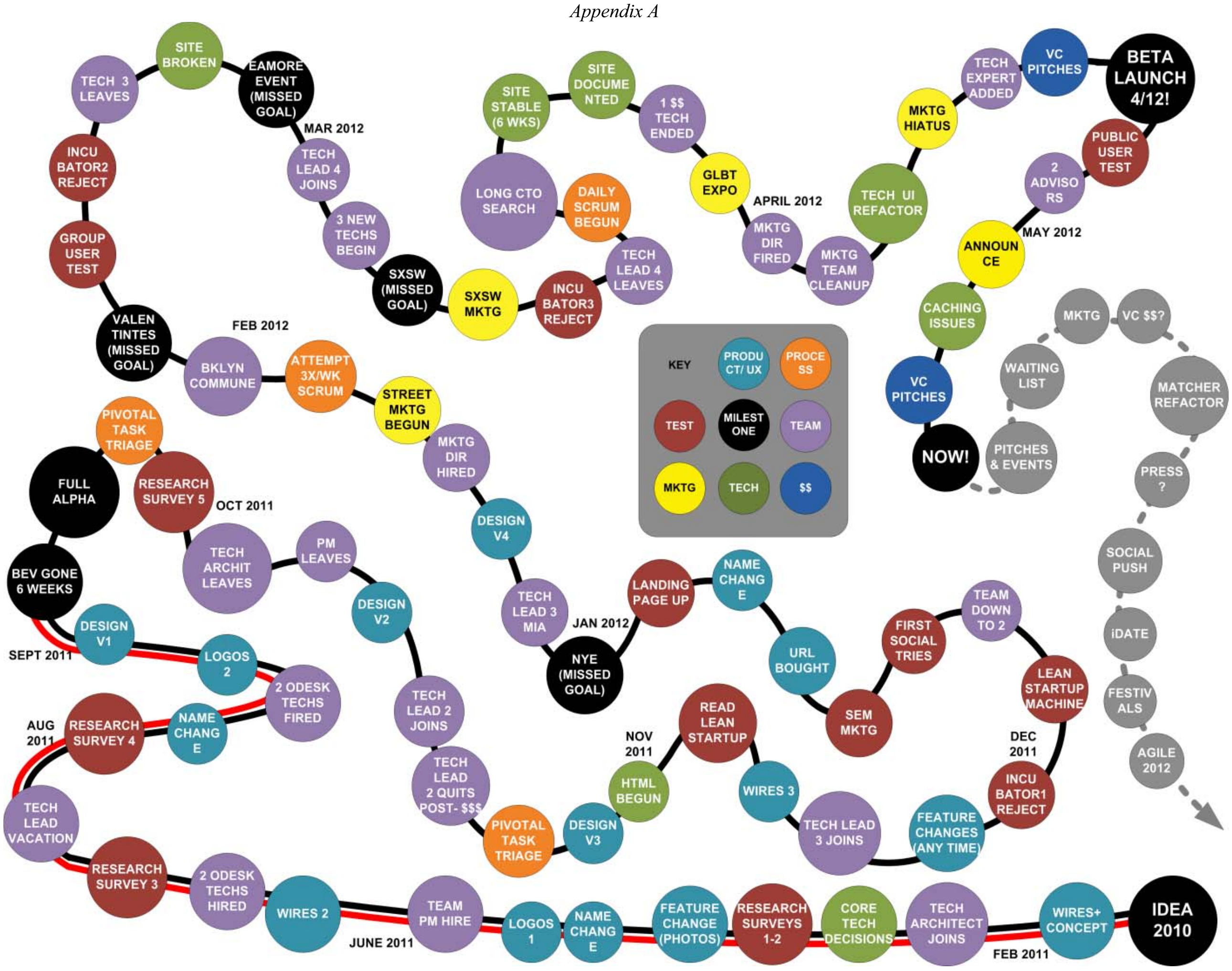
[13] Ries, Eric. The Lean Startup. New York: Crown Business, 2011, p. 43.

[14] Ries, Eric. The Lean Startup. New York: Crown Business, 2011, p. 99.

[15] Aronoff, David. "VC Confidential". Retrieved from SkillShare class: http://www.skillshare.com/VC-Confidential-anything-you-want-to-know-about-raising-capital/2123840005/528824230 , April 2012.

[16] Foundups, "About". Retrieved from: http://foundups.com/about Foundups.com, May 2012.

Appendix A


IDEA 2010
WIRES+ CONCEPT
FEB 2011
TECH ARCHITECT JOINS
CORE TECH DECISIONS
RESEARCH SURVEYS 1-2
FEATURE CHANGE (PHOTOS)
NAME CHANG E
LOGOS 1
TEAM PM HIRE
JUNE 2011
WIRES 2
2 ODESK TECHS HIRED
RESEARCH SURVEY 3
TECH LEAD VACATION
AUG 2011
RESEARCH SURVEY 4
NAME CHANG E
2 ODESK TECHS FIRED
LOGOS 2
DESIGN V1
SEPT 2011
BEV GONE 6 WEEKS
FULL ALPHA
PIVOTAL TASK TRIAGE
RESEARCH SURVEY 5
OCT 2011
TECH ARCHIT LEAVES
PM LEAVES
DESIGN V2
TECH LEAD 2 JOINS
TECH LEAD 2 QUITS POST- $$$
PIVOTAL TASK TRIAGE
DESIGN V3
HTML BEGUN
NOV 2011
READ LEAN STARTUP
WIRES 3
TECH LEAD 3 JOINS
FEATURE CHANGES (ANY TIME)
INCU BATOR1 REJECT
DEC 2011
LEAN STARTUP MACHINE
TEAM DOWN TO 2
FIRST SOCIAL TRIES
SEM MKTG
URL BOUGHT
NAME CHANG E
LANDING PAGE UP
NYE (MISSED GOAL)
JAN 2012
TECH LEAD 3 MIA
DESIGN V4
MKTG DIR HIRED
STREET MKTG BEGUN
ATTEMPT 3X/WK SCRUM
BKLYN COMMUNE
FEB 2012
VALEN TINTES (MISSED GOAL)
GROUP USER TEST
INCU BATOR2 REJECT
TECH 3 LEAVES
SITE BROKEN
EAMORE EVENT (MISSED GOAL)
MAR 2012
TECH LEAD 4 JOINS
3 NEW TECHS BEGIN
SXSW (MISSED GOAL)
SXSW MKTG
INCU BATOR3 REJECT
TECH LEAD 4 LEAVES
DAILY SCRUM BEGUN
LONG CTO SEARCH
SITE STABLE (6 WKS)
SITE DOCUME NTED
1 $$ TECH ENDED
GLBT EXPO
APRIL 2012
MKTG DIR FIRED
MKTG TEAM CLEANUP
TECH UI REFACTOR
MKTG HIATUS
TECH EXPERT ADDED
VC PITCHES
BETA LAUNCH 4/12!
PUBLIC USER TEST
2 ADVISO RS
MAY 2012
ANNOUN CE
CACHING ISSUES
VC PITCHES
NOW!
PITCHES & EVENTS
WAITING LIST
MKTG
VC $$?
MATCHER REFACTOR
PRESS ?
SOCIAL PUSH
iDATE
FESTIV ALS
AGILE 2012
KEY
PRODU CT/ UX
PROCE SS
TEST
MILEST ONE
TEAM
MKTG
TECH
$$