Fail fast, fail often, use the cloud

Bruce Scharlau
University of Aberdeen

You think you know about startups

You've read stories about startups that grew from nothing to massive successes:

eBay, Amazon, Twitter, and Instagram

The little business that grows miraculously

These businesses seem to grow and grow no matter what happens

The little company always seems to do the right thing

We never see mistakes until much later after the books have been written and the movies are made

We never see the mistakes

Surely they must've made mistakes?

We should see & acknowledge the mistakes

To learn more about startups, the process and using the technology behind startups we need to look closer and see the mistakes: especially if we're thinking of our own startup

Game one: word at a time story

What do you do when you don't know what's happening next?

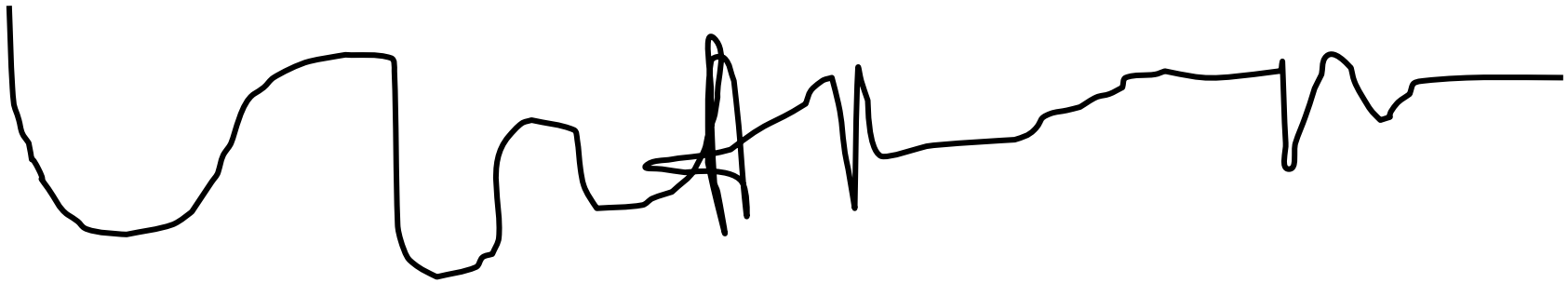
Game idea from Keith Johnson

1: We should acknowledge mistakes as early as possible

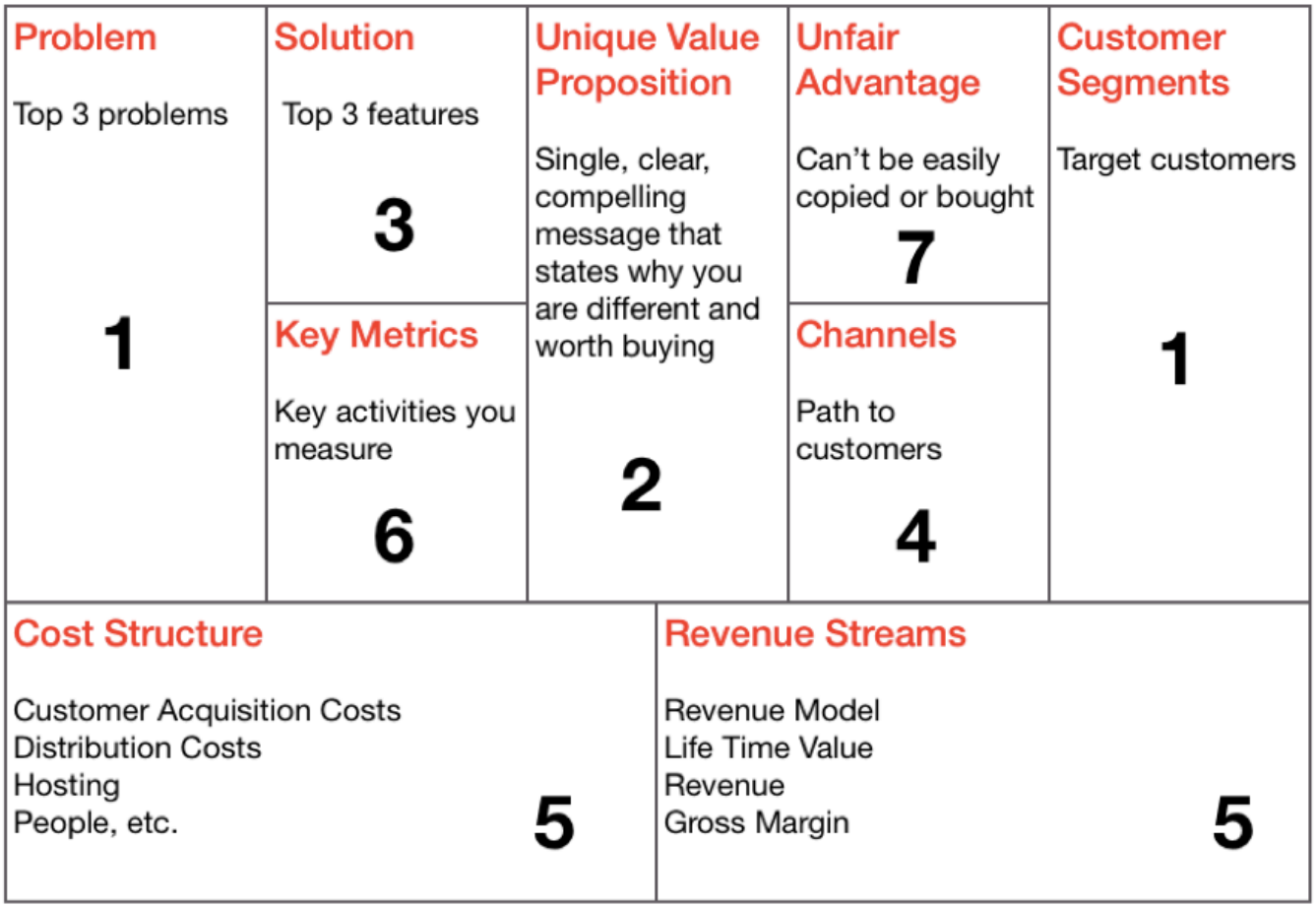
We need to incorporate feedback about errors into our work: the earlier the better

We assume this:

But experience this:



Use lean canvas to validate assumptions

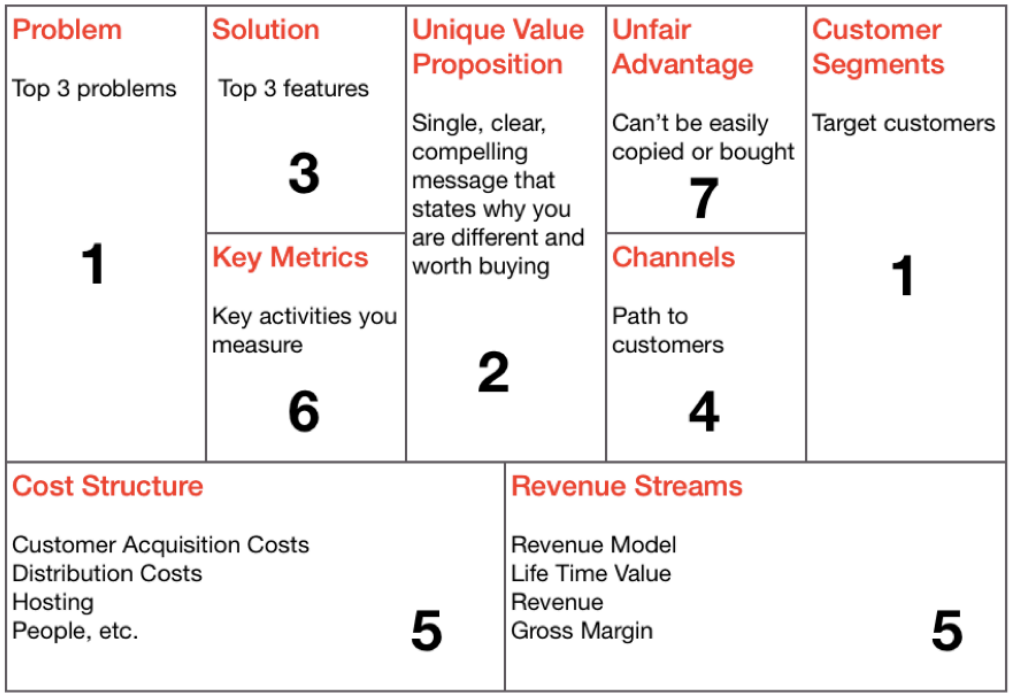


http://leanstack.com/LeanCanvas.pdf

Use this before you build anything

Lean canvas matches problems to specific customers

Find people with problems you can understand and then build solutions. Discuss problems with people in cafes, discuss on line.



Lean canvas focuses on unique

You need an unfair advantage not easily replicated to create high cost of entry for others. This could be proprietary tools, or use design to create unique service.

You can’t compete on price though. That never works.

http://neildavidson.com/download/dont-just-roll-the-dice/

Service design lets you co-create solutions with users

Work with potential clients/customers to develop solutions to their problems because they know more than you do and they will always surprise you so that you can find your unique proposition if you're listening.

Use service design tools to prototype ideas

Use tools to gather information which show what people do, and not what they ‘say they would do’. Believe actions.

http://www.servicedesigntools.org/repository

Service design lets you see opportunities for improvement

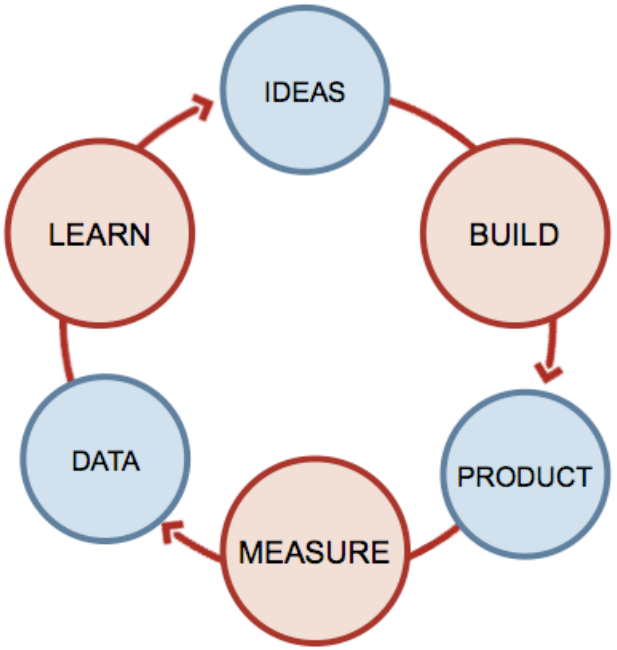
Service design is the glue that holds the other parts of a product and its components together and is why you go into one café for the experience and ignore another café.

Eliminate mistakes sooner in order to spend less

The sooner you find out where you went wrong the sooner you can fix your mistaken assumption and head in the right direction

Finding problems early leads to new discoveries

Validated learning using a build->measure->learn cycle means you find the unknowns you didn't know about, and can take advantage of them

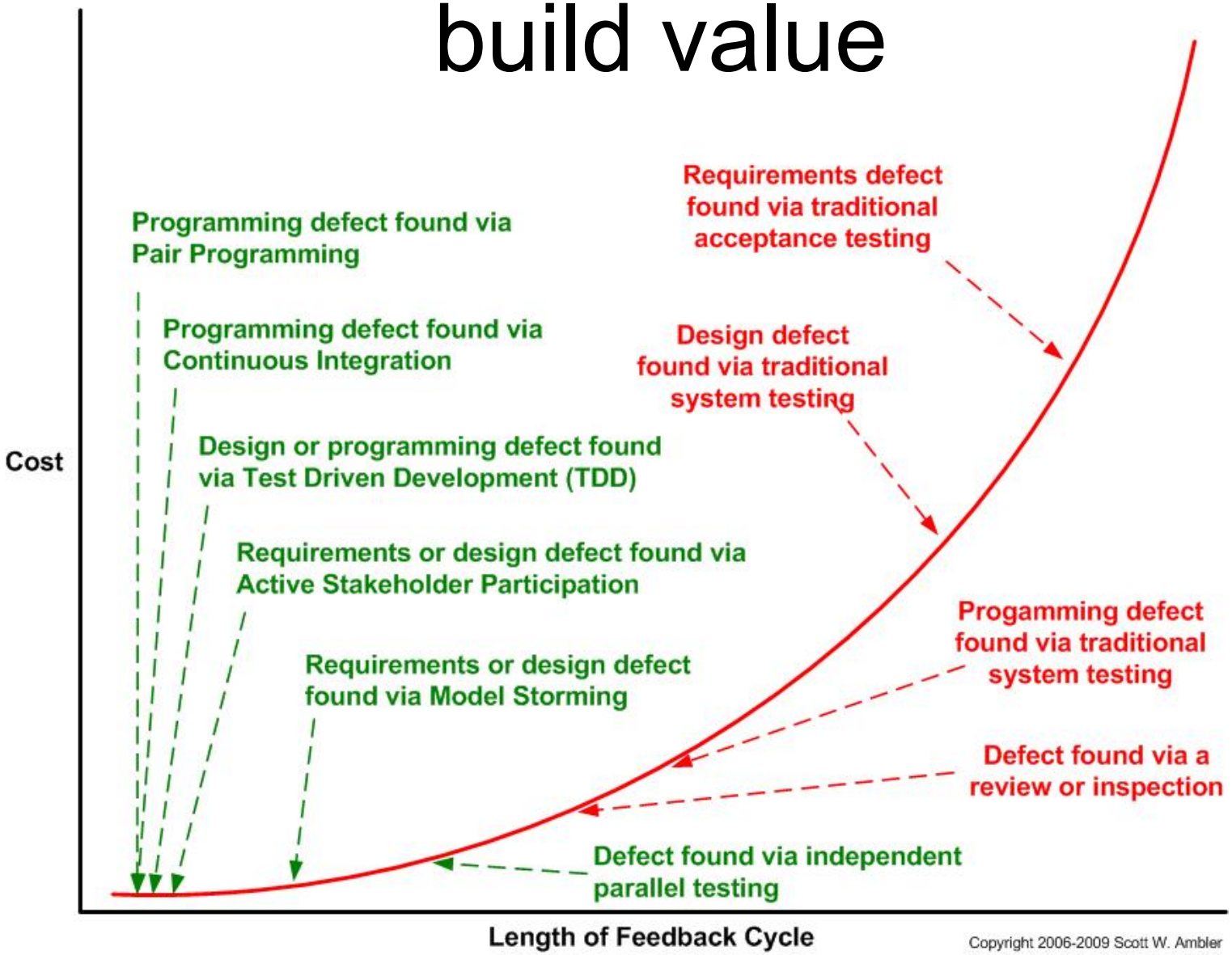


http://lean.st

http://theleanstartup.com/principles

Finding problems early means we build value

http://www.agilemodeling.com/essays/costOfChange.htm



Game two: The hand game

Team wins when have everyone holding different fingers up, eg one, two, three, four, five

Game idea from Keith Johnson

2: We learn more from failure than success

If you never made a mistake how would you know what led to your success? Was it the service, the price, the market? Right place at right time? You wouldn't know, you couldn't repeat this, or enable others to learn

http://www.amazon.co.uk/Managing-Design-Factory-Product-Developers/dp/0684839911/

Recoverable failure is ok

Spend what you could happily lose, and be able to undo your deployments so that you can be no worse off than you were before, apart from what was spent

http://www.amazon.co.uk/Principles-Software-Engineering-Management-Gilb/dp/0201192462/

Cloud platforms provide space to play with ideas

Play provides a safe environment to experiment were mistakes are ok

Play is fun and provides the fastest way to learn something

https://www.heroku.com

http://aws.amazon.com

https://www.google.co.uk

https://bitbucket.org

http://www.cloudfoundry.com

Cloud platforms enable easy deployment

Often upload from within development environment like Visual Studio, Eclipse, or via a push from your source control system such as git

We should see small experiments

There should be a/b testing, there should be trial sites for different markets, there should be experiments where you want to know what the ‘correct/better’ option might be

Cloud platforms enable affordable trials

This is all inexpensive compared to what we learn. Yes, we have to spend some time and money doing these experiments, but we learn about the idea, customers, and possible solutions

Cloud platforms enable discussions

We can create forums and surveys or sites to engage with potential customers

Cloud platforms enable easy adaption

Using virtual machines of the cloud we can quickly and easily try new components such as noSQL, message systems and other tools to improve our service as we find a need to pivot our initial business idea

Cloud platforms enable scaling

We can scale up and down using tools to grow our service.

We are not stuck having to decide when we should buy another server and possibly waste our money, or miss an opportunity if we wait too long.

3: Incremental and iterative development enables early validation

By using agile development we can plan how to release the minimal product/service to test our ideas as soon as possible

http://www.infoq.com/minibooks/scrum-xp-from-the-trenches

Incremental enables affordable learning

We don’t spend as much if we have small slices and also can bring in more money as these help increase revenue flow

Incremental enables progress

We can start from where we are with incremental changes. We don't have to start with a clean slate. We can use what's to hand to validate assumptions before we spend lots on an unproven idea

Incremental enables measurable progress towards goals

By using small slices we can see how much each experiment brings us closer to our business goals

http://impactmapping.org

Cloud platforms enable MVP

Build minimum viable product and no more

http://www.amazon.co.uk/Lean-Startup-Eric-Ries/dp/0307887898

Cloud platforms enable growth from ‘balloon’ to MVP in stages

Float a balloon and go from there via the developing conversation one increment at a time

Cloud platforms enable simple to complex

Find a platform to suit your needs at the time and don't be afraid to change platforms if/ when it's necessary

Game three: A multi-tasking game

Three Projects

A B C D E F G H I

1, 2, 3, 4, 5, 6, 7, 8, 9, 10

i ii iii iv v vi vii viii ix x

Two Ways

1. Serially?

   A B C D …

2. Concurrently?

   A 1 i B …

How much of your capacity is wasted?

5% 10% 20% 30% 40% ????

Thanks to @clarkeching who suggested the idea

UNIVERSITY OF ABERDEEN

Waiting for whole loses revenue and opportunity

You don’t need all features before you release your service. Others will get there ahead of you.

Better to release MVP and have revenue start immediately as break even sooner this way

Agile development means you can earn while you learn

Start with MVP and keep on with build->measure->learn and use experiments to learn more about your customers as you grow/improve your app

Agile development increases revenue with lower costs and earlier return

Using agile means you don’t dig as deep a hole before you breakeven on your investment

! Acknowledge mistakes as learning opportunities

By looking for errors, wrong assumptions and failing, we learn and develop opportunities to take advantage of them

You'll always make mistakes

You have forgotten something important about whatever you're developing and the sooner you find out the sooner you can take advantage of the new learning

Learning one error provides space to find the next one

There are always errors in your assumptions and your job is to find enough of them to develop an income before you run out of money

Incorporate ‘learning opportunities’ in order to succeed

By using lean, service design and agile approaches you can fail fast. You can use the cloud systems to provide the means to deploy these easily and grow a business

Fail to succeed through the cloud

Bruce Scharlau
University of Aberdeen
MSc Software Entrepreneurship
http://yourspinout.com
@scharlau
b.scharlau@abdn.ac.uk