

# How do you know that your product works?

Colombo Agile Conf, June 2014

Consultant



**Henrik Kniberg**

henrik.kniberg@crisp.se  
@HenrikKniberg

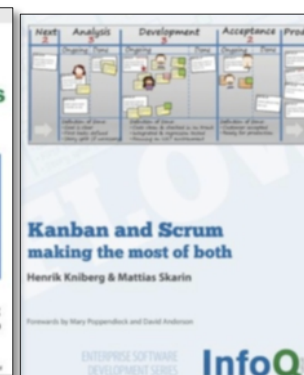
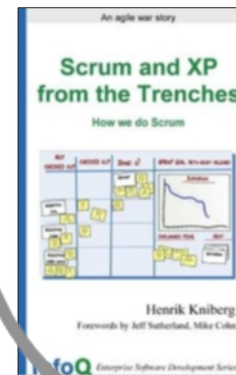
Father

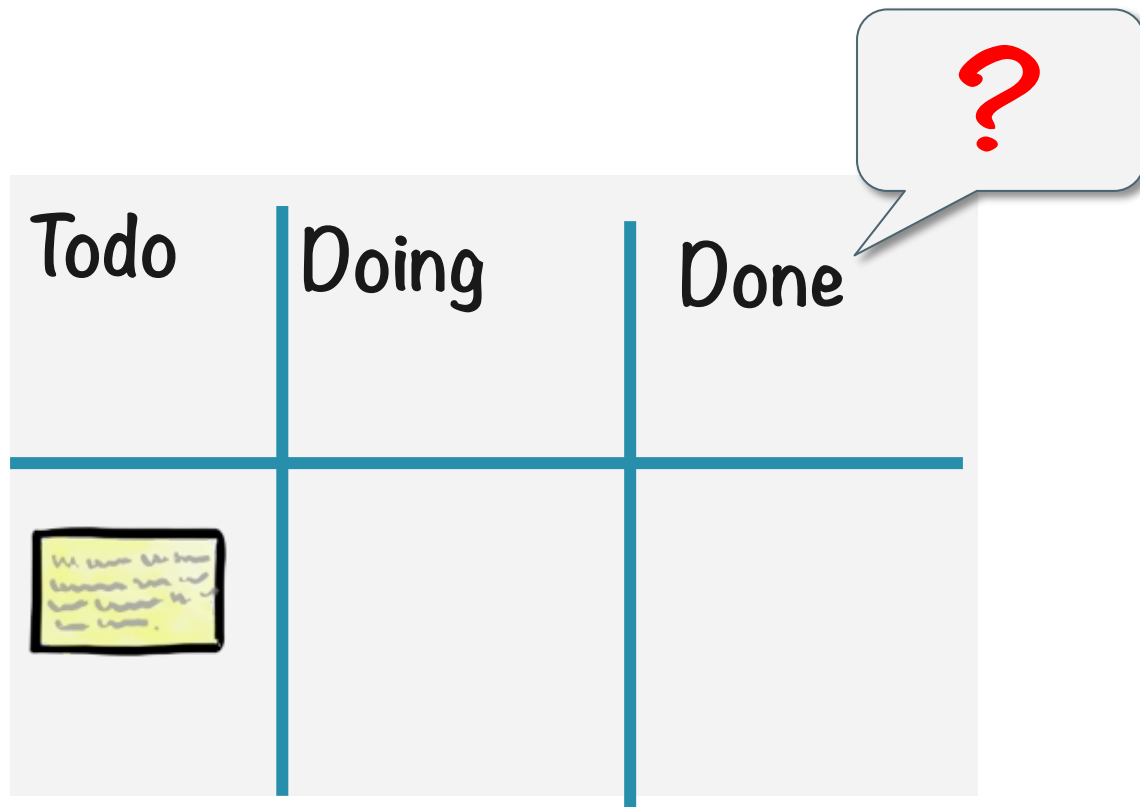


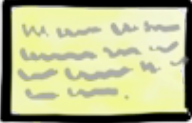
Agile & Lean coach



Author





Todo	Doing	Code committed
		



Developer

I'm done!

Where's my product?



User



Where's my product?

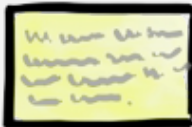


User

We're Done!



Developer

Code committed	Tested	In production
		

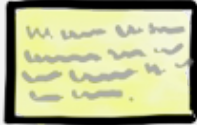
100% completed requirements  
No reported defects



We're Done!



Wait... is anyone actually using it?

Code committed	Tested	In production	People actually use it
			

100% completed requirements  
No reported defects

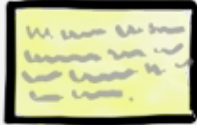


NOW we're surely done!

Aren't we?

<http://www.youtube.com/watch?v=FWTtcOQjOXI>

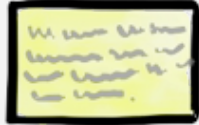


Code committed	Tested	In production	People actually use it
			

100% completed requirements  
No reported defects

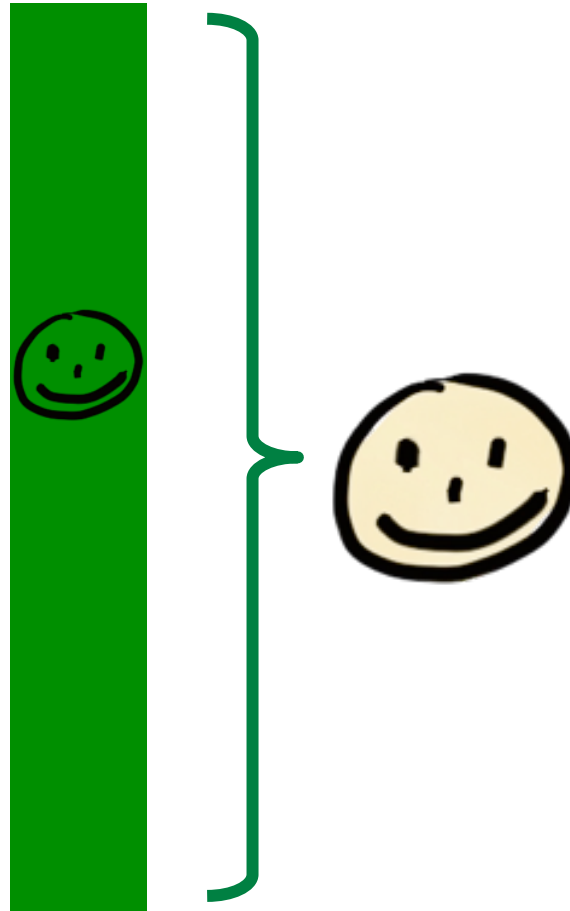




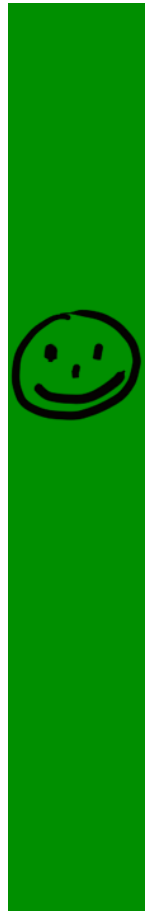
Code committed	Tested	In production	People actually use it	Solves the user's problem
				

100% completed requirements  
No reported defects





Value of  
your solution



Value of  
your solution



Value of old  
solution



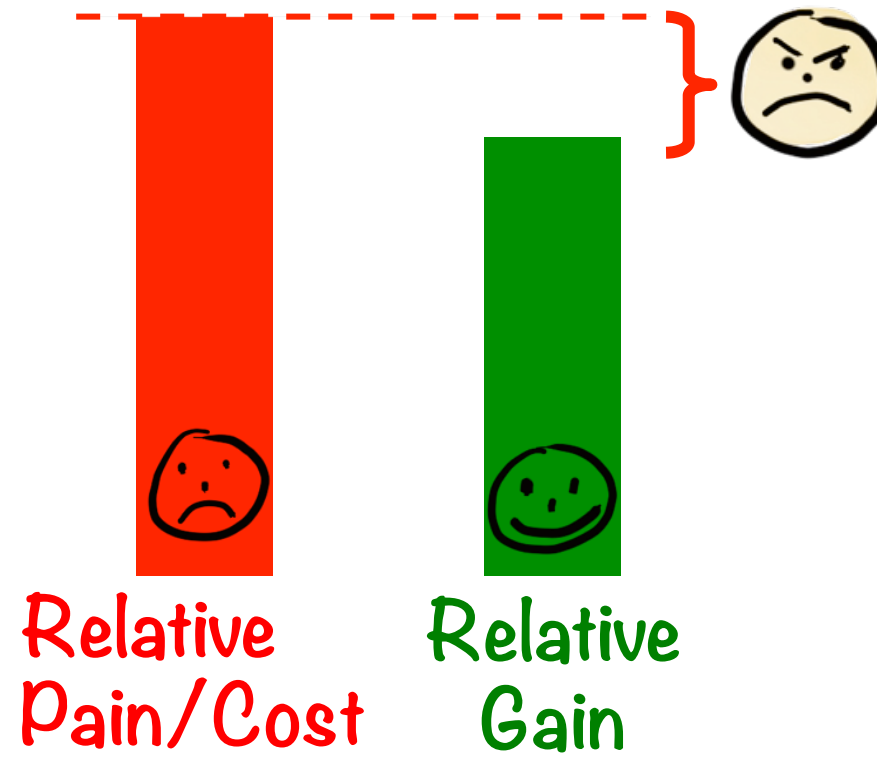


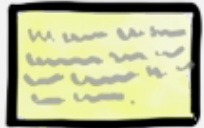
Relative gain

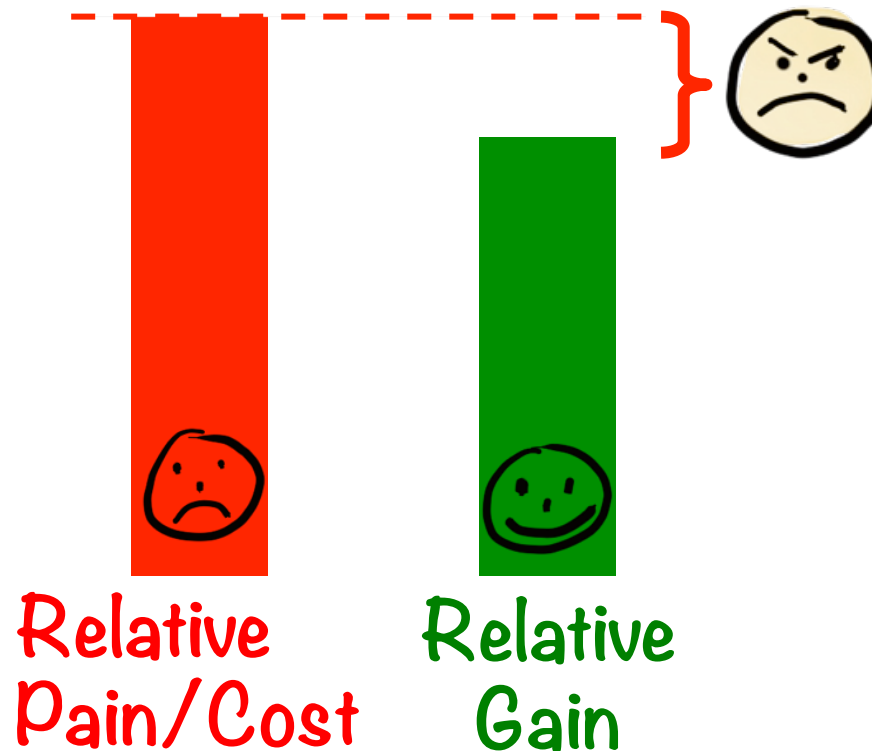


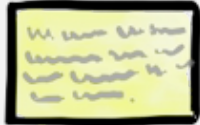
Value of  
your solution

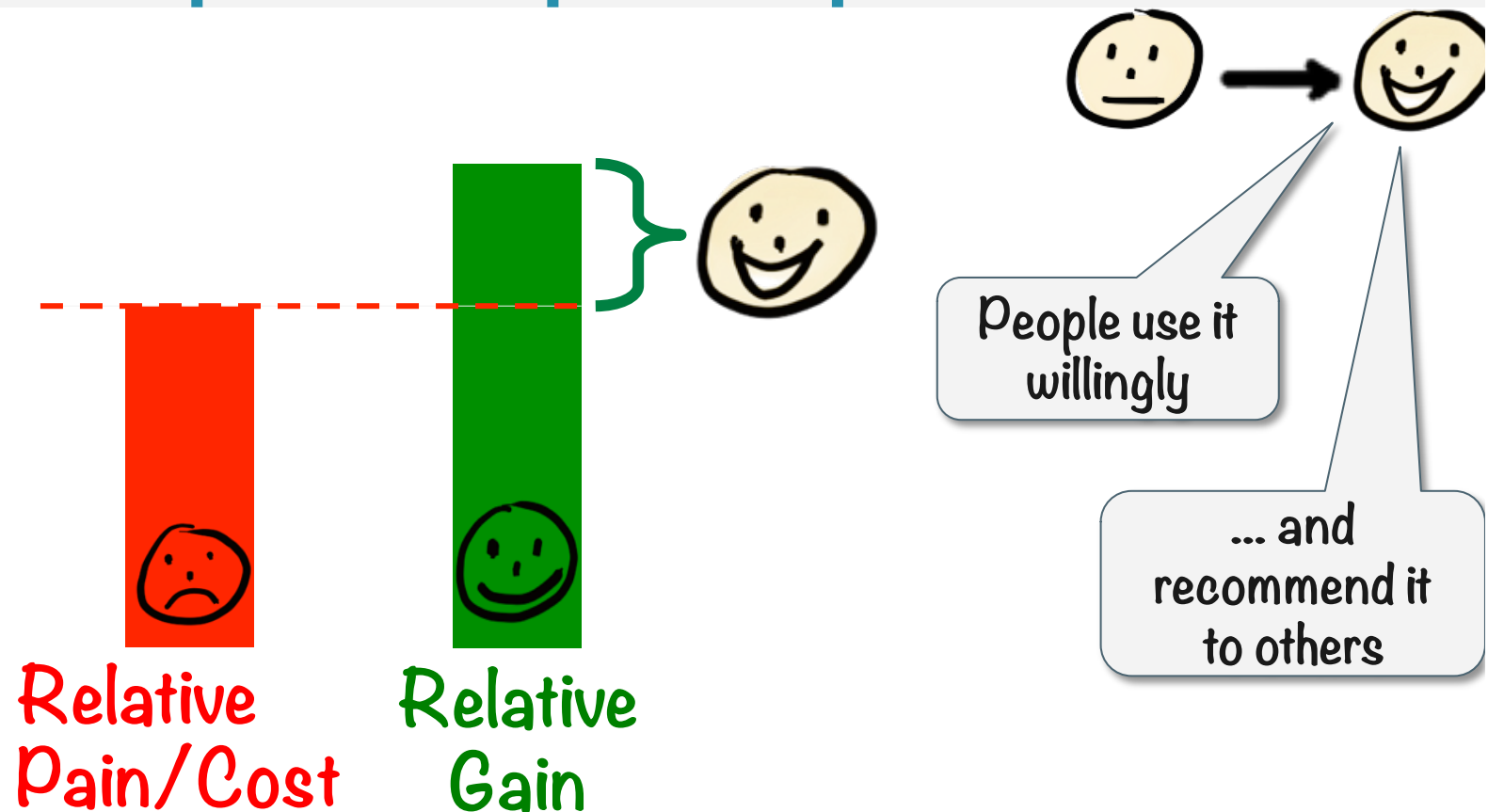
Value of old  
solution



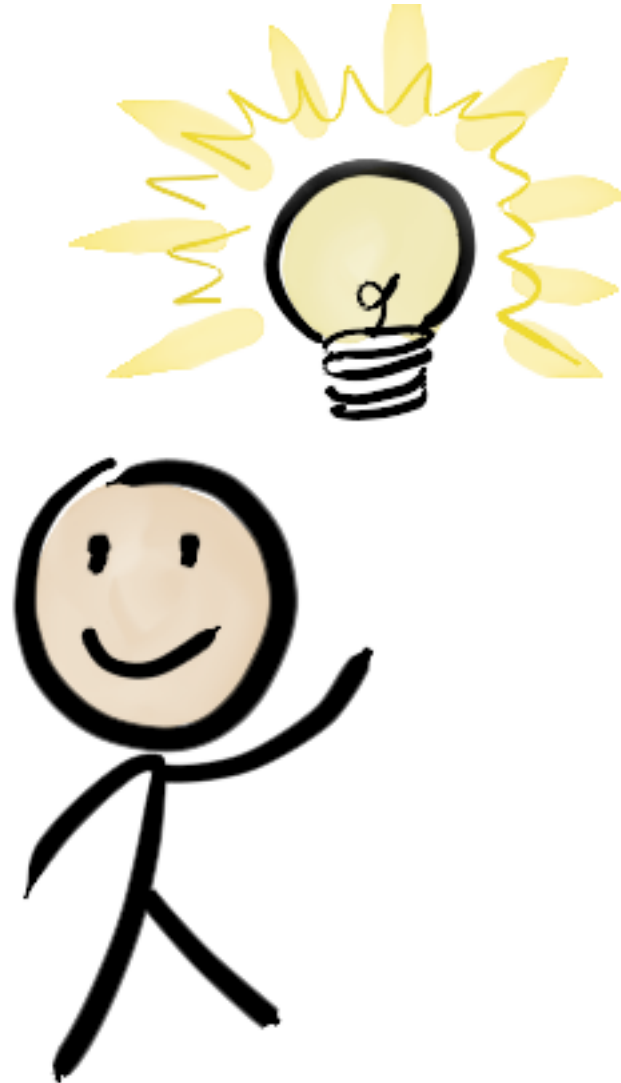
Code committed	Tested	In production	People actually use it	Solves the user's problem	... in a way that is better than before
					



Code committed	Tested	In production	People actually use it	Solves the user's problem	... in a way that is better than before
					



# All products start with a Great Idea!





# Risk



Business risk



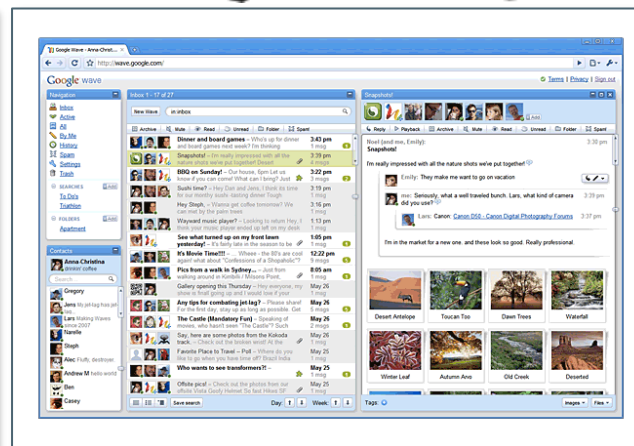
Technical risk



Social risk



Cost & schedule risk



# Assume you are building the **WRONG** thing!

Suppose we're  
building the wrong  
thing....

How can we find out  
as quickly as  
possible?



- List your hypotheses
- Build an MVP to validate/invalidate it
  - "Minimum Viable Product" – the fastest & cheapest thing you can build to field-test your hypothesis

# Example: Dropbox

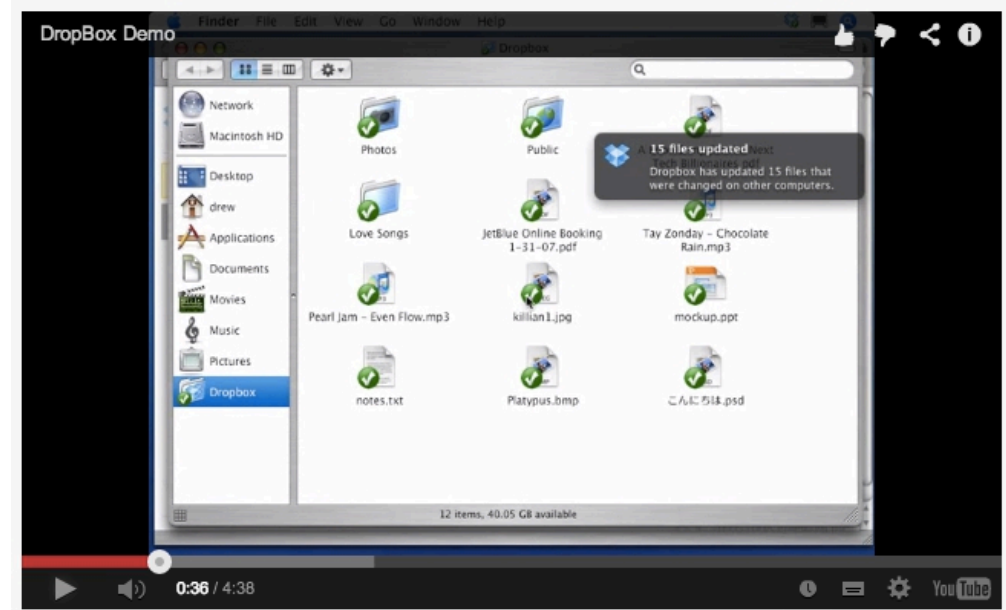
## Hypotheses

1. File sync is a problem for people

2. Our product will solve the problem

3. People will want to use our product

## Demo video



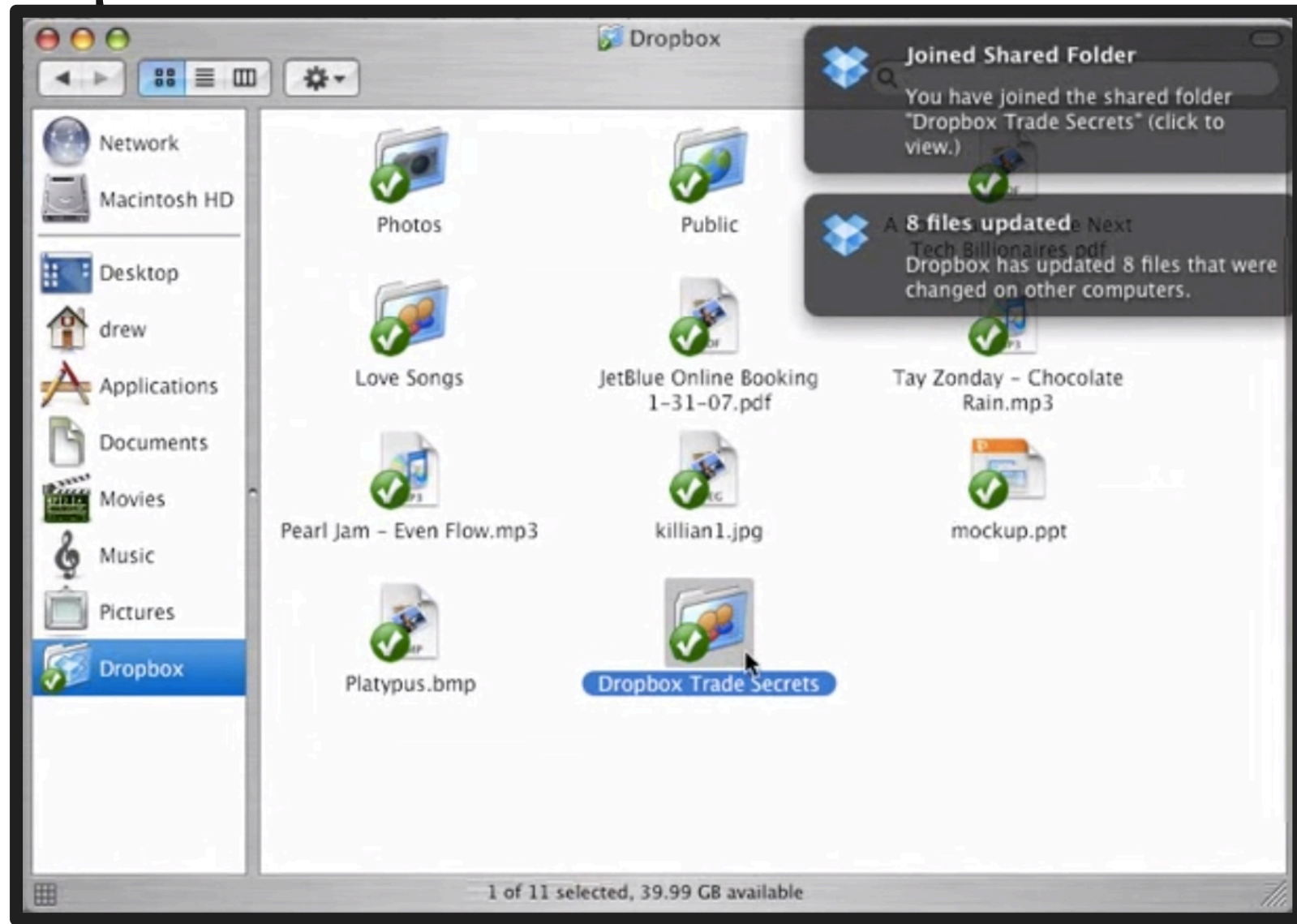


# Dropbox

[www.getdropbox.com](http://www.getdropbox.com)

Drew Houston  
[beta@getdropbox.com](mailto:beta@getdropbox.com)

# Dropbox MVP





# Dropbox MVP



It drove hundreds of thousands of people to the website.

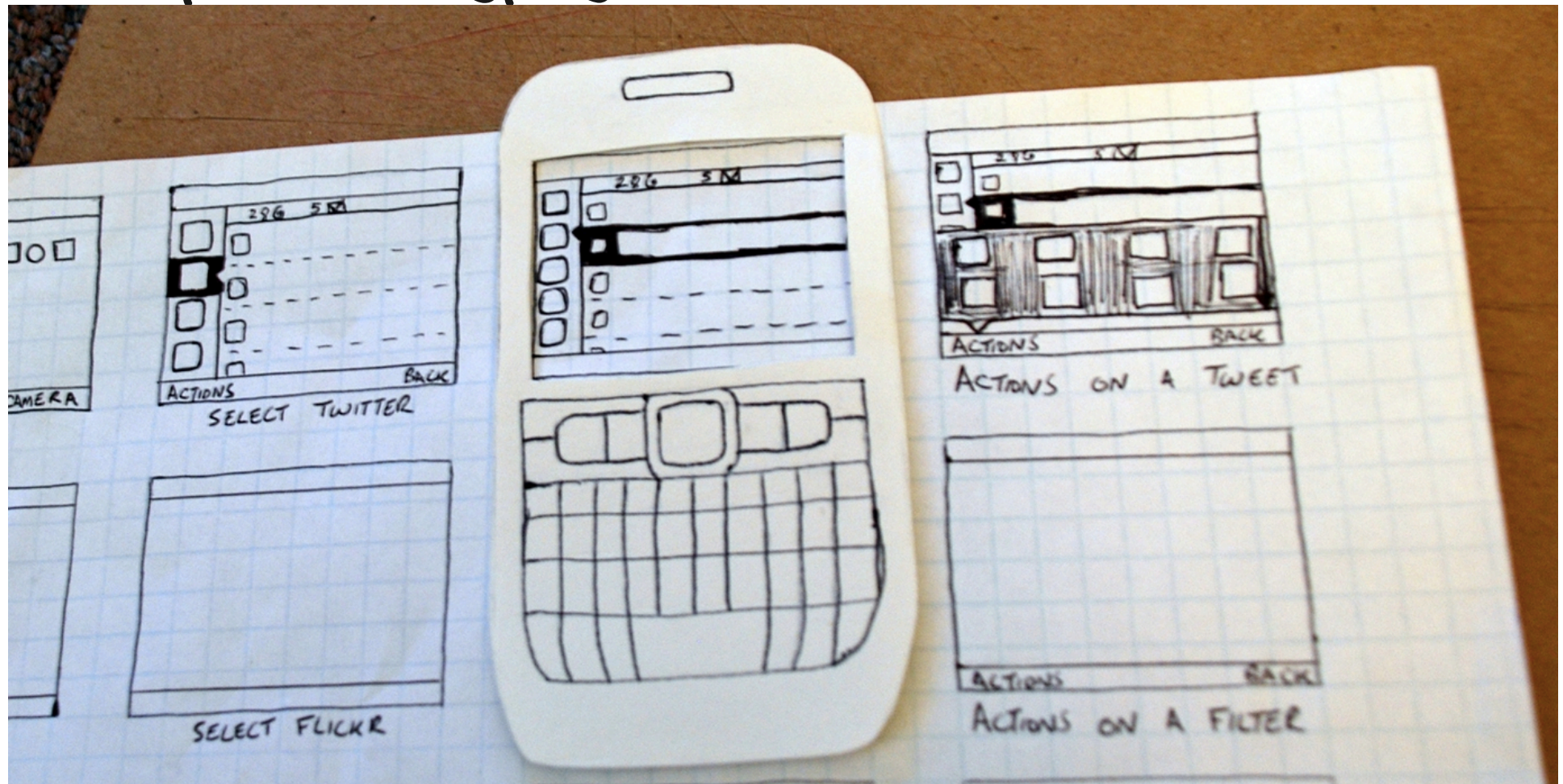
Our beta waiting list went from 5,000 people to 75,000 people literally overnight.

It totally blew us away.



Drew

# Paper Prototyping = Lo-tech MVP



Credit: Ariel Waldman, on Interaction Design/ Rachel Ilan  
<http://chiefdisruptionofficer.com/helpful-rapid-prototyping-methods-and-tools-to-bring-digital-ideas-to-life-fast/>



# Pirate metrics



AARRRR!

**A**quisition

Do people  
come?

**A**ctivation

Do they use the  
product?

**R**etention

Do they come  
back?

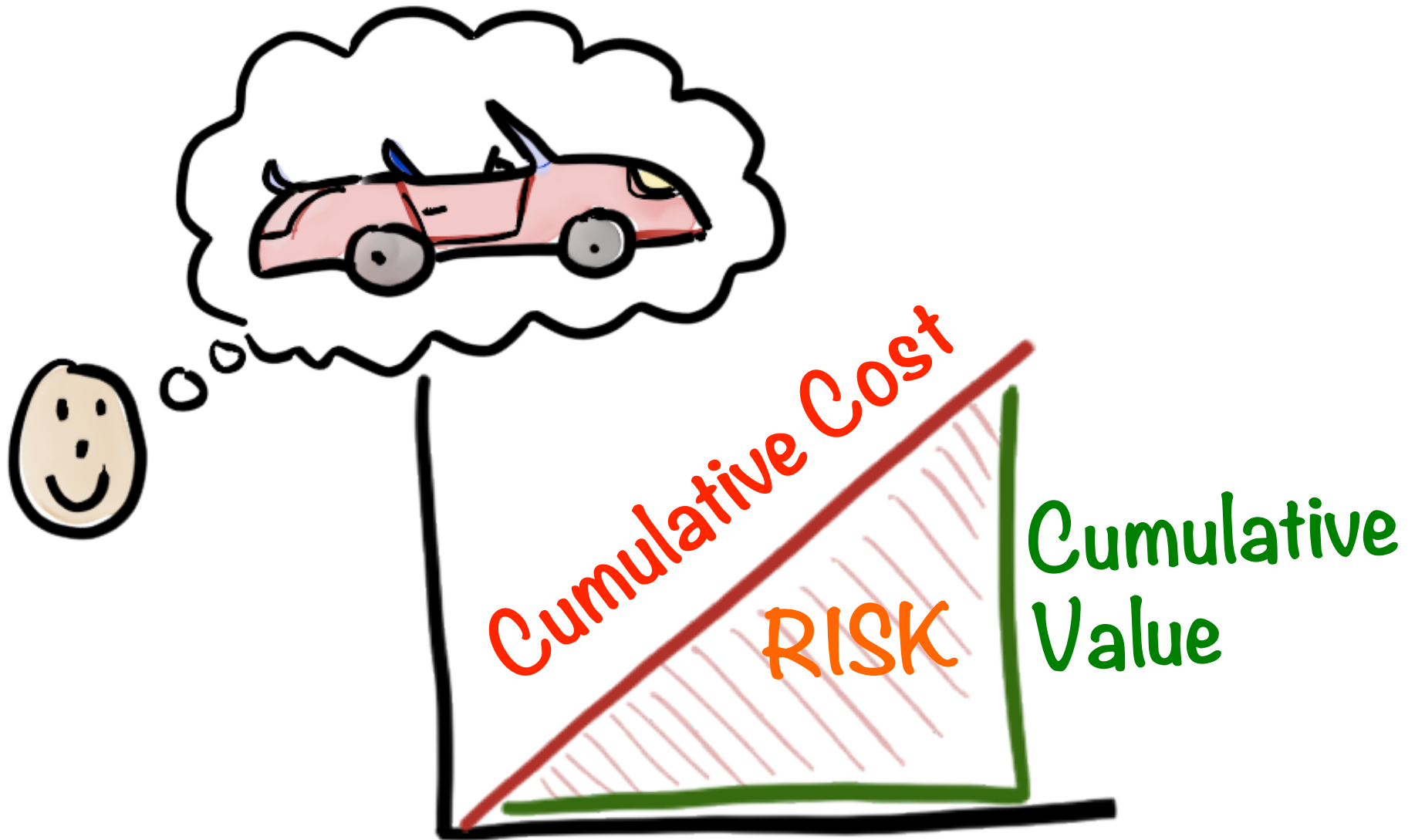
**R**eferral

Do they recommend  
it to others?

**R**evenue

Do they pay?

Big Bang = Big Risk

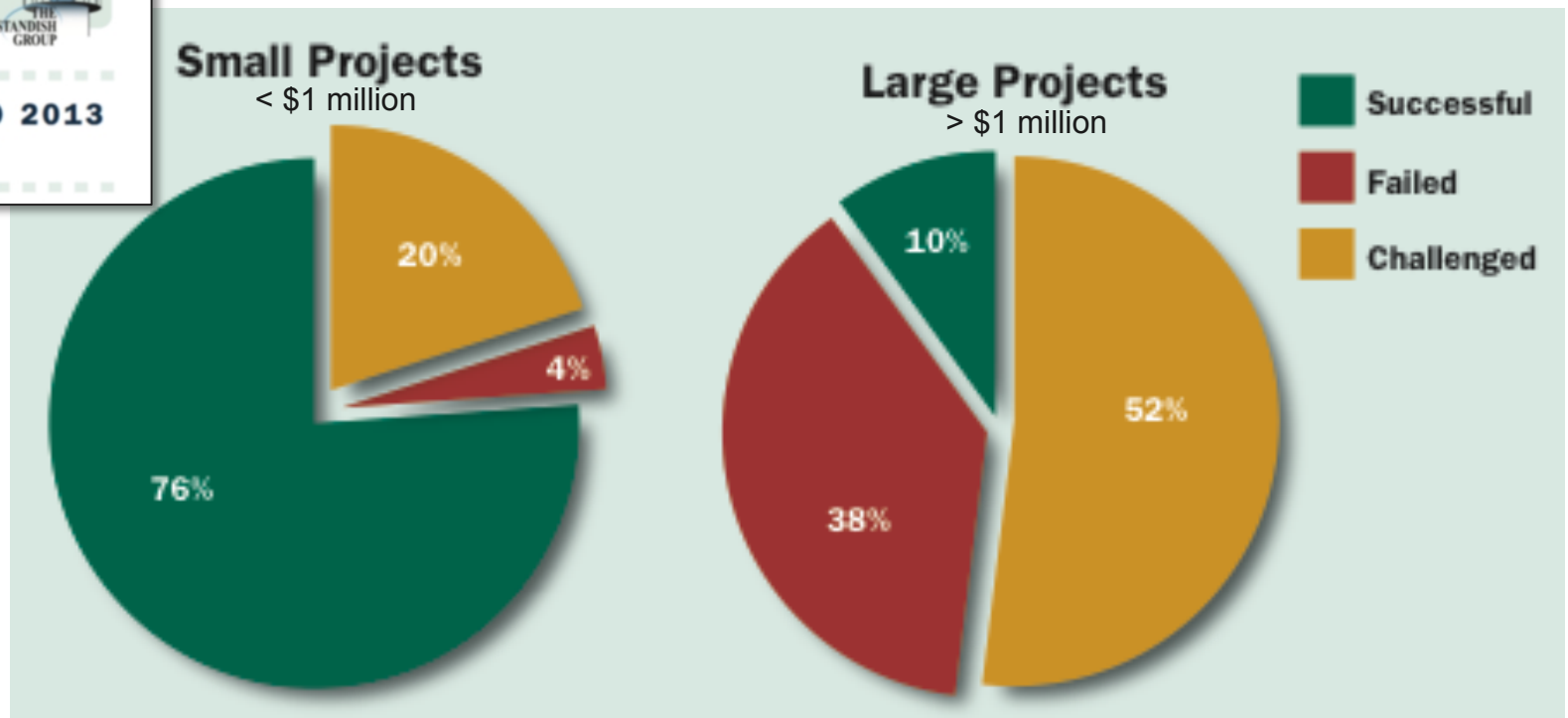


# Big Projects usually fail. Regardless of process.



"The secret to project success is to **strongly recommend and enforce limits on size and complexity.**"

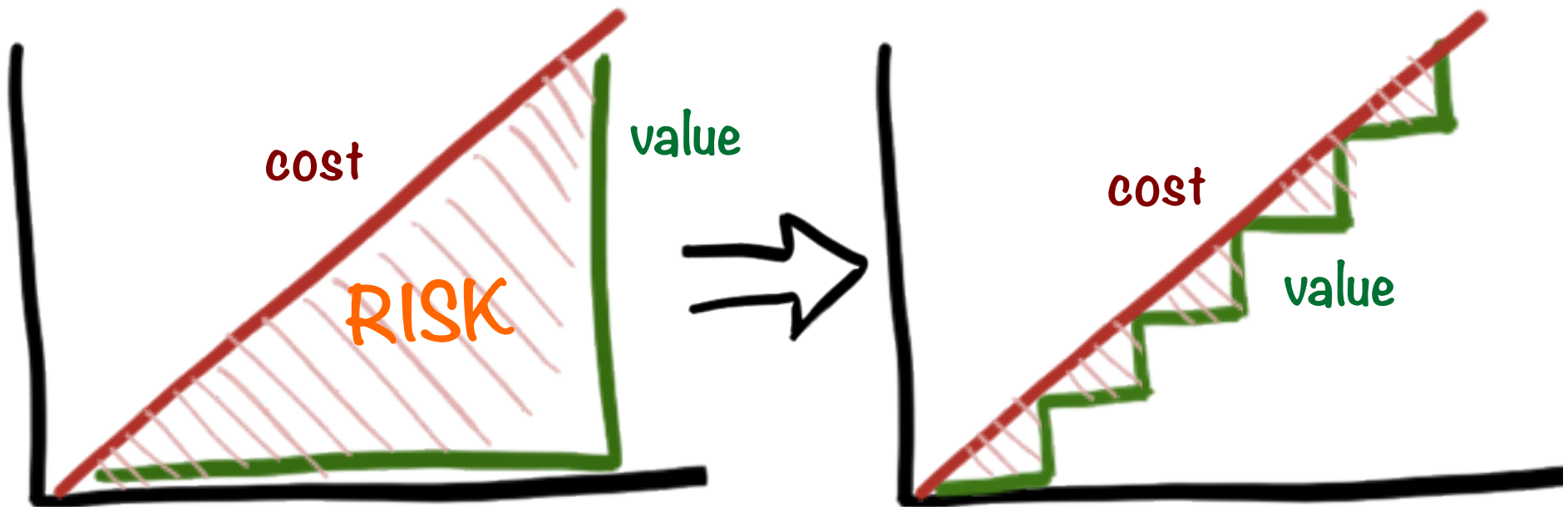
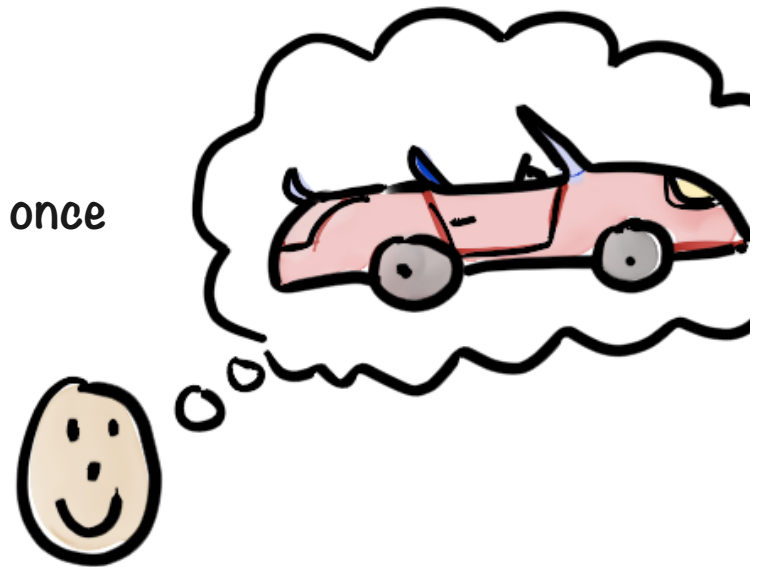
"These two factors trump all other factors."



# Agile = Iterative + Incremental

Don't try to get it all right from the beginning

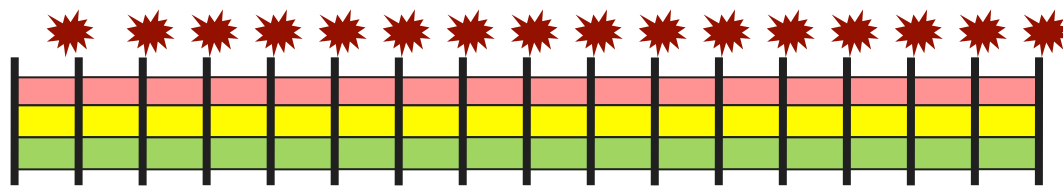
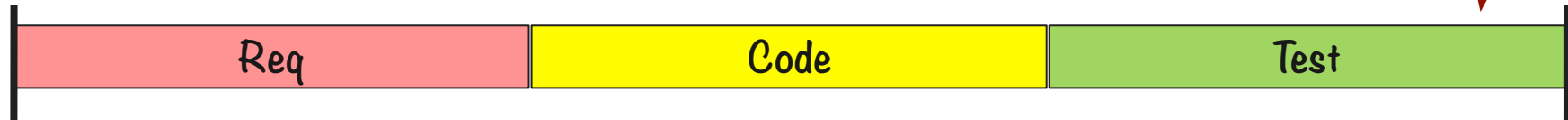
Don't build it all at once



# Release must be REALLY easy!

Release = Drama!

Release!

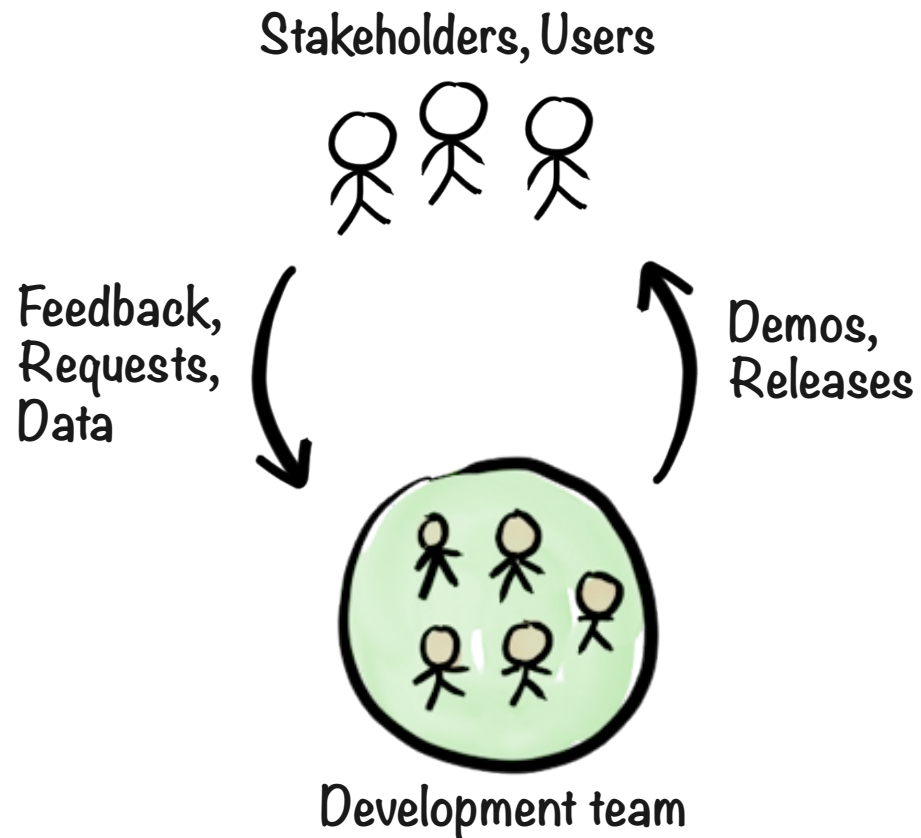


Release = Routine

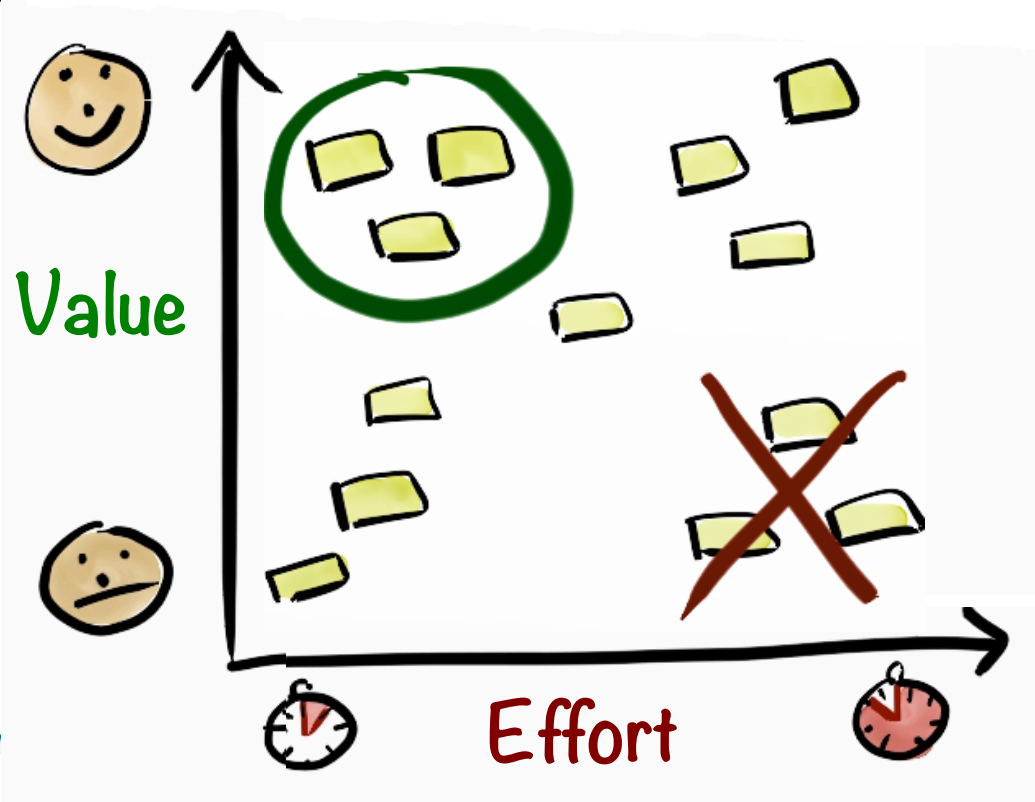
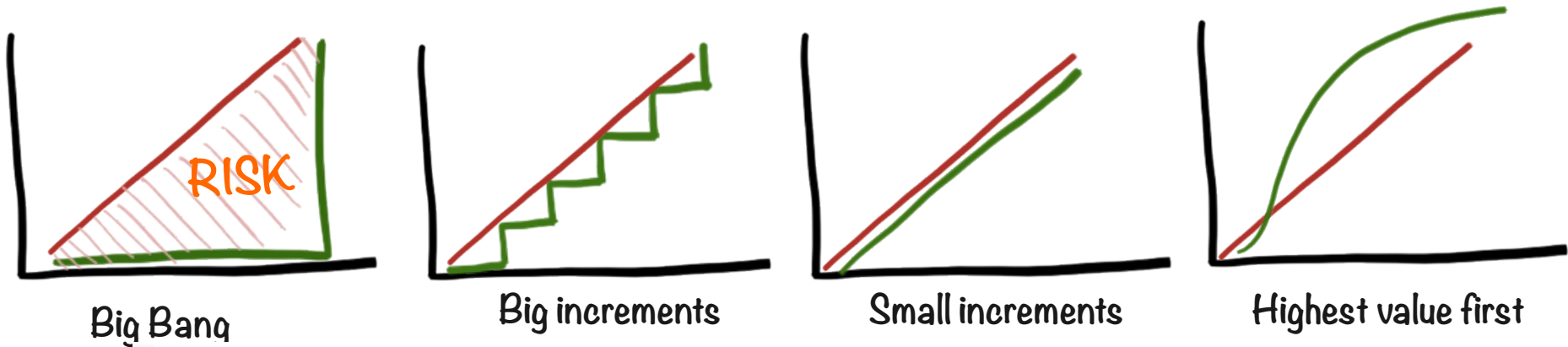


# Fastest learner wins!

Delivery frequency = Speed of learning



# Improving the Value Curve



Not like this....



1



2



3



4

---

Like this!



1



2



3



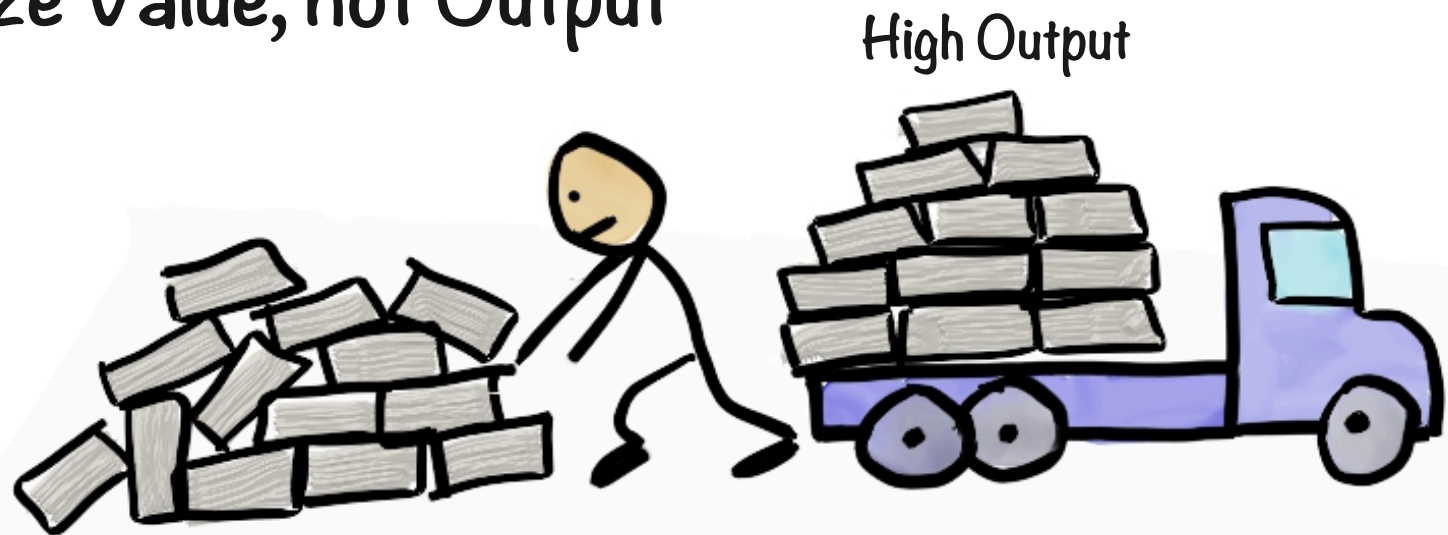
4

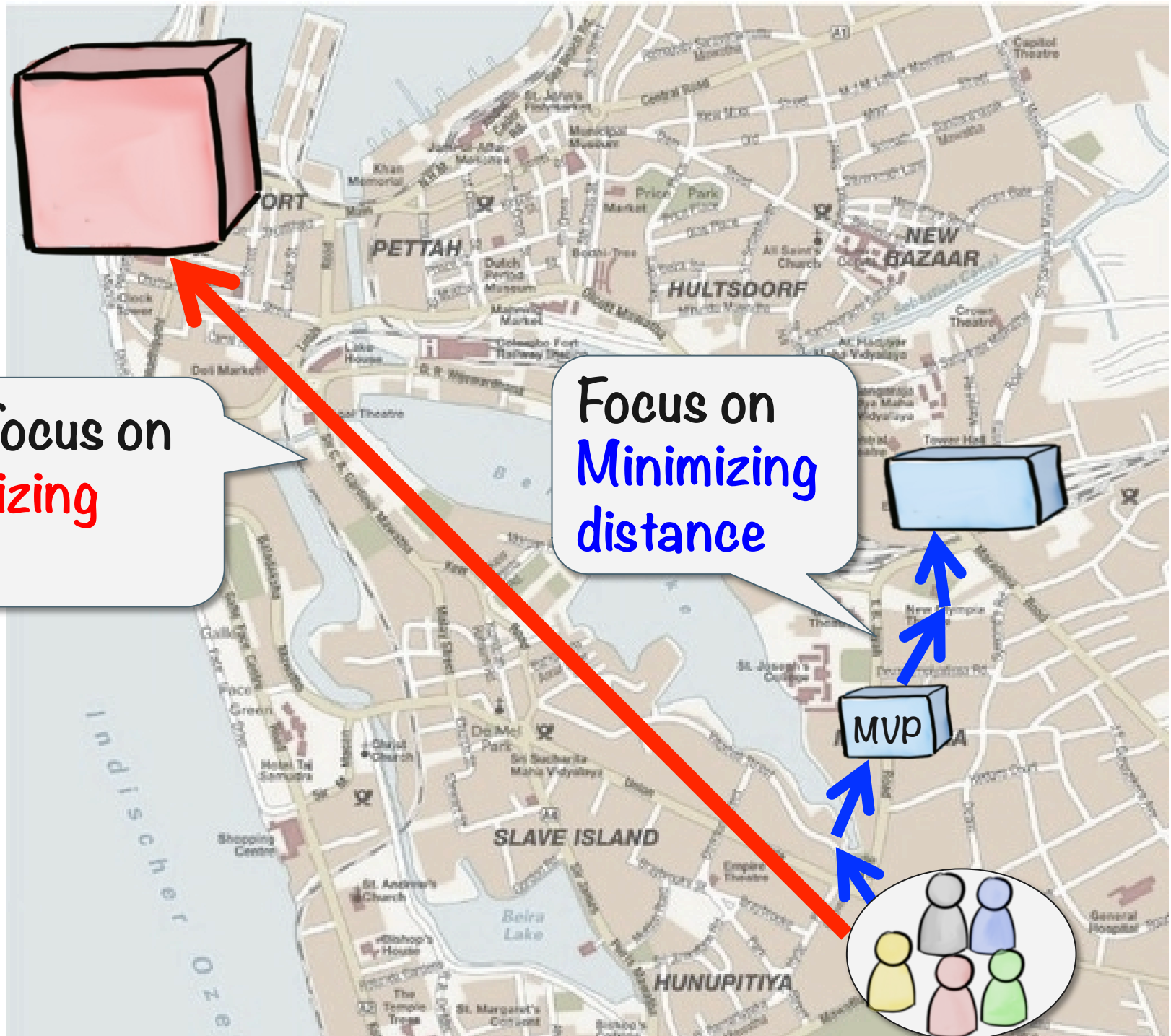


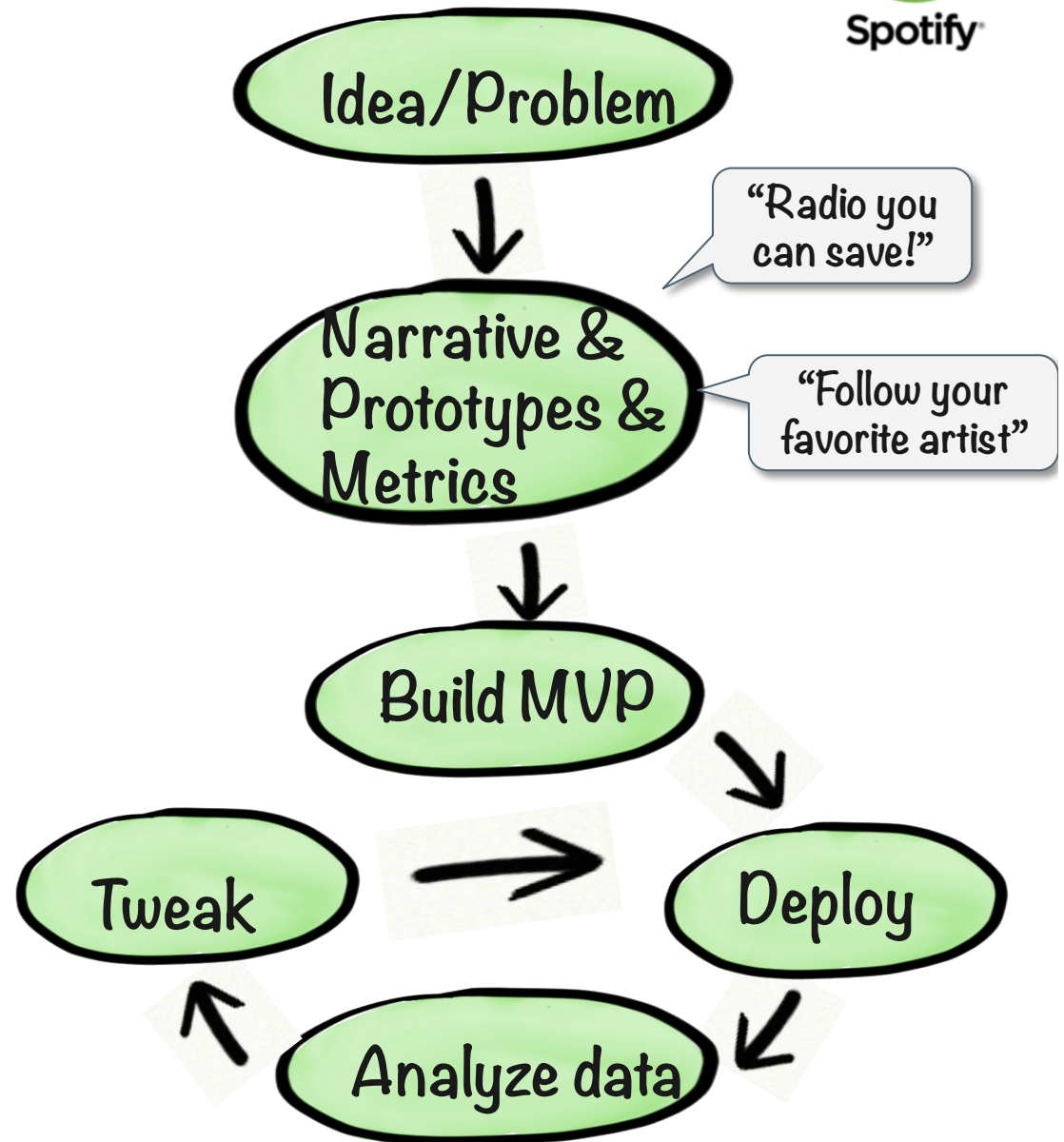
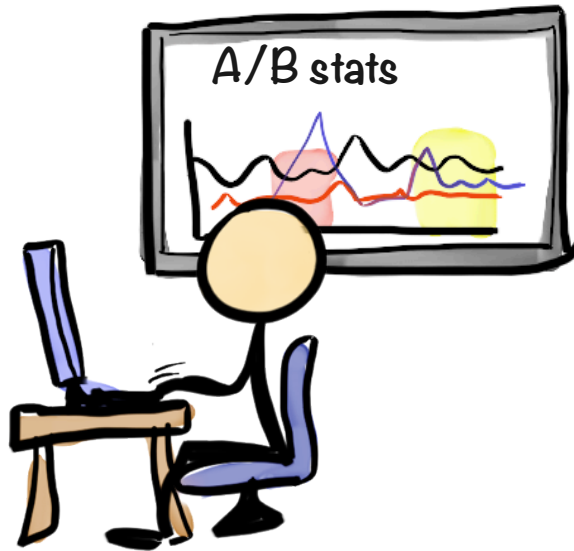
5



# Maximize Value, not Output









# Manifesto for Agile Software Development

We are uncovering better ways of developing software by doing it and helping others do it.  
Through this work we have come to value:

Individuals and interactions over processes and tools  
Working **Product** over comprehensive documentation  
**User** collaboration over contract negotiation  
Responding to **Feedback** over following a plan

That is, while there is value in the items on the right, we value the items on the left more.

# How do you <sup>ensure</sup>~~know~~ that your product works?

## 1. Understand the problem

Who are the stakeholders?

What need do they have, that we want to solve?

How will we know when we've solved it?

How will we know if we're moving in the right direction?

## 2. Iterate until you've solved it

Minimize the distance to MVP

Deliver, measure, adjust continuously